

AUSGABE



9

AMIGA

Markt & Technik

DM 16,-

ÖS 120,- / Sfr 16,-
Lit 16000 / hfl 21,- / dkr 75,-

SONDERHEFT

Die besten

LISTINGS

des Jahres

20 Spitzenprogramme

zum Sparpreis

**Tolle Spiele ■ Faszinierende Grafikprogramme
■ Nützliche Anwendungen ■ Hilfreiche Tools**

WWW.HOMECOMPUTERWORLD.COM

AMIGA

Markt & Technik

SOFTWARE
EXTRA

Software zum Taschengeldpreis

Jedes Programm nur DM 49,-*

(sFr 45,- / 65 490,-)

Grafik



Amiga Extra Nr. 1: Grafik I
Drei Programme, die die außergewöhnlichen Grafikfähigkeiten des Amiga nutzen!
Bestell-Nr. 38708



Amiga Extra Nr. 4: Grafik II
Spielerisch Bobs erzeugen. IFF-Bilder werden zu Bobs und Images. Generieren von animierten Icons.
Booster: par excellence.
Bestell-Nr. 38725

Spiele



Amiga Extra Nr. 12: Spiele
Highway 42: Als Kurier auf dem Planeten Cervezia. Warlords: Ein Brettspiel für taktisch geschickte Spieler.
Zargon: Joystick-Action.
Bestell-Nr. 38769



Amiga Extra Nr. 5: Spiele
Breaking out: Actionspiel mit toller Grafik und Sound.
Decoder: Verwandeln Sie Ihren Amiga in eine Morsestation.
Megamind.
Bestell-Nr. 38752



Amiga Extra Nr. 3: Spiele
Bliff: Eine ausgeklügelte Variante des Billards.
Quadrige: Ein Spiel für Denker, angelehnt an «Vier gewinnt».
Wiking: ein Strategiespiel.
Bestell-Nr. 38724



Amiga Extra Nr. 13: Regnum
Regnum ist Ihr Königreich in einer imaginären Welt. Die Amiga-Maus dient Ihnen als Zepter. Ihre Aufgabe: Sichernt und vergrößern Sie Ihren Besitz.
Bestell-Nr. 38781

Musik



Amiga Extra Nr. 6: AudioWorx
Ihr privates Sampling-Studio.
Bestell-Nr. 38748



Amiga Extra Nr. 9: Sonix-Hitliste
Tolle, in Sonix editierbare und digitalisierte Geräusche und Effekte für eigene Musikstücke.
Bestell-Nr. 38753



Amiga Extra Nr. 2: Disk-Utilities I
Disk-Ed V6, Select Copy, D Copy II, Check, Bootgiri Plus. Mit einem Super-Diskeditor.
Bestell-Nr. 38726



Amiga Extra Nr. 10: Disk-Utilities II
DIMO: Diskettenmonitor. Recover II: Datenrettung.
TUC - The Ultimate Cruncher: Dateien extrem komprimieren.
Bestell-Nr. 38766



Amiga Extra Nr. 14: MenuMind 1.0
MenuMind bietet Ihnen eine menügesteuerte, grafische Programmierungsumgebung, die den CLI völlig ersetzt.
Bestell-Nr. 38771



Amiga Extra Nr. 11: Kartekasten
Finden Sie sich in Ihrem Diskettenbestand nicht mehr zurecht? Oder, oder, oder. Mit Dateiverwaltungen werden Ihre Probleme gelöst.
Bestell-Nr. 38768

Spielend lernen!



Amiga Extra Nr. 7: Erdkunde I
Die Reihe «Spielend lernen» verknüpft das Begeisterte des Amiga mit dem Nützlichen. Stupide Paukerie wird durch Kurzweil ersetzt.
Bestell-Nr. 38774



Amiga Extra Nr. 8: Englisch I
«Englisch I» vermittelt Ihnen Grundkenntnisse der englischen Sprache.
Bestell-Nr. 38775



Amiga Extra Nr. 16: Erdkunde II
Spielend lernen! Vereinigte Staaten von Amerika. Interaktives Lernprogramm für alle ab 12 Jahren, mit Übungskurs und Quiz.
Bestell-Nr. 38776



Amiga Extra Nr. 17: Mathematik I (Geometrie)
«Mathematik I» vermittelt Ihnen die Grundlagen der Geometrie bis hin zur sphärischen Trigonometrie.
Bestell-Nr. 38777



Amiga Extra Nr. 18: Mathematik II (Algebra)
vermittelt Ihnen die Grundlagen der Algebra, die die Voraussetzung für die gesamte weiterführende Mathematik sind.
Bestell-Nr. 38778



Amiga Extra Nr. 19: Physik I
Grundlagen der Mechanik, der Wärmelehre und der Optik. Animationen und Soundeffekte erhöhen die Verständlichkeit und die Lerneffizienz.
Bestell-Nr. 38779

INFO-COUPON

Bitte senden Sie mir Ihr Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software

Name

Straße

PLZ/Ort

Bitte ausschneiden und schicken an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Peter-Straße 2, 8013 Haar bei München

Ami SH 9

*Unverbindliche Preisempfehlung

Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser

Markt & Technik

Zeitschriften · Bücher
Software · Schulung

WWW.HOMECOMPUTER.MAGAZINE

Software kompakt

● Ein prall gefülltes Software-Paket – und nur Spitzenprogramme. Wer wünscht sich das nicht?

Für Sie haben wir die besten Listings des Jahres 1989 zusammengestellt. Viele »Programme des Monats« vom AMIGA-Magazin und die »besten Stücke« aus den Sonderheften sind dabei.

● Was hat uns dazu bewogen?

Nun, zu Weihnachten bekamen Tausende einen neuen Amiga. Die Freude war sicher groß, doch besonders zu Anfang fehlt es an guter Software. Dem wollen wir abhelfen. Aus den verschiedensten Bereichen haben wir Spitzen-Listings zusammengestellt: faszinierende Grafik-Software, unterhaltsame Spiele, nützliche Anwendungen und hilfreiche Tools, die einem den Computeralltag erleichtern. Für jeden ist etwas dabei. Stöbern Sie nur in diesem Software-Paket, es lohnt sich.

● Noch ein weiterer Punkt bewog uns zu dieser Software-Sammlung. Gingen Ihnen auch schon die folgenden Gedanken durch den Kopf? »Mensch, das Programm des Monats

hat mich in dieser Ausgabe unwahrscheinlich interessiert. Aber dafür gleich die Programmservice-Diskette kaufen – nein.« Und zum Abtippen hatte man weder Lust noch Zeit. Also ließ man es bleiben. Für mehr als 20 dieser Programm-Renner gibt man gern das Geld für die Beschaffung der Programmservice-Diskette aus.

● Warum dann noch die Listings im Heft? Wir müssen auch an diejenigen Leser denken, bei denen der Geldbeutel sehr eng geschnürt ist.

● Auch diese Amiga-Fans sollen die Chance erhalten, in den Genuß der Software zu kommen – und meist steht Ihnen etwas mehr Zeit zur Verfügung.

● Natürlich interessiert uns Ihre Meinung zu dieser Sammlung. Schreiben Sie uns, an die »Redaktion Sonderheft«. Wir warten mit Spannung auf Ihre Reaktionen.

Also, viel Spaß beim Fliegen mit dem Computer, Entwickeln von Riesen-Sprites, Tischtennis-Kampf gegen den elektronischen Gegner, Fiebern an der Börse und, und, und...



Ihr
Gottfried Knechtel
(Stellv. Chefredakteur)

Gottfried Knechtel

Grafik

6 Sprite-Editor

Überdimensionale Sprites spielend leicht entwickeln. Der »Amiga-Objekt-Editor« bedeutet das Ende mühevoller Kleinarbeit.

18 Der Konvertierer

Mit »AOE-Convert« bauen Sie die mit dem Amiga-Objekt-Editor entwickelten Super-Sprites in Ihre Basic-Programme ein.

21 Zauberei mit Bildern

»IFF-Filter« modifiziert Ihre Farbbilder auf faszinierende Weise.

27 Grafik in der dritten Dimension

Lassen Sie sich faszinieren von dem Spiel mit Licht und Schatten, das Sie mit diesem »Ray Tracer« konstruieren können.

30 Bewegung in 3 Dimensionen

Gittergrafiken – bewegt, perspektivisch, dreidimensional. Kein Problem für »3DAmim«.

Spiele

40 Nur Fliegen ist schöner

Vom Starten bis zu Kunstflugmanövern über herrlicher Landschaft. Lernen Sie Ihre Maschine zu beherrschen mit diesem kurzen aber leistungsstarken Flugsimulator.

50 Ping-Pong pur

Tückische Angaben, Schmettern und Bälle anschneiden. Wer schafft es, den spielstarken Computer zu besiegen?

55 Krieg der Sterne

»Fantasy« entführt Sie weit in die Zukunft. Planeten kämpfen um die Vorherrschaft.

61 Trainingslager

Trainieren Sie Ihr Gehirn. Ein Denkspiel der Spitzenklasse.

67 Spiel, Spaß & Spannung

»Buh Au CS« ist ein zweites Strategiespiel, das nach dem Spielprinzip von »Eimerketten« neu entwickelt wurde.

76 Börsenfieber am Amiga

Spekulationsgewinne und Husarenstreiche – »Broker« bringt die Faszination der Börse in Ihr Wohnzimmer.

Tools

84 Packen wir's

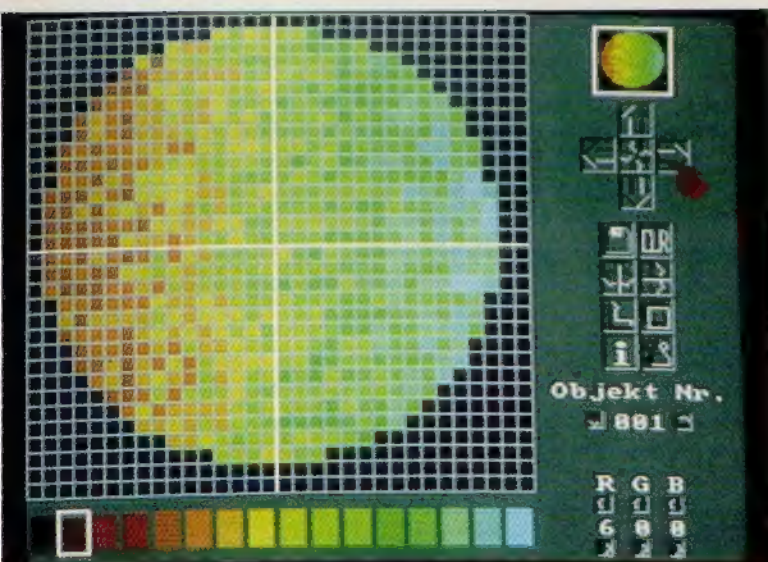
Viele Programme mehr auf einer Diskette speichern: »MasterCruncher« packt alles. Bis über 50% werden die Dateien kürzer.

90 Der Entseuchungstrupp

Ein Virus-Killer, den jeder Amiga-Besitzer braucht. Er schützt Ihren Computer vor Link- und Bootblock-Viren und kontrolliert das gesamte System.

100 Geräteinspektion

Wollen Sie wissen, ob ein externes Laufwerk oder eine Festplatte angeschlossen ist? »TestDev.h« untersucht, welches Device wirklich vorhanden ist.



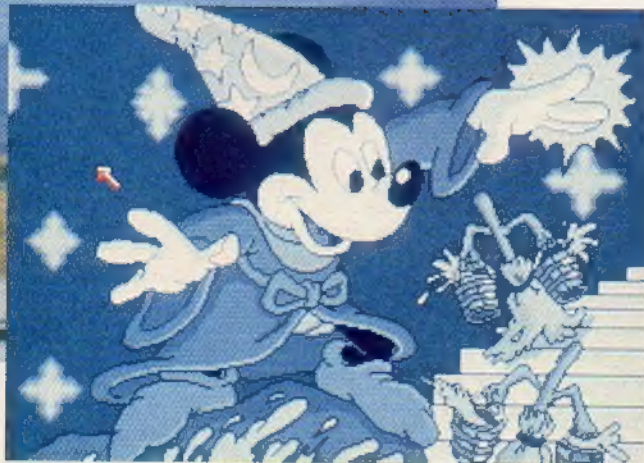
Mit dem mausgesteuerten »Amiga-Objekt-Editor« lassen sich Riesen-Sprites und Animationssequenzen schnell und mühelos erstellen. Ein Programm, das jeder Grafik-Fan braucht.

SEITE 6



Wollten Sie schon immer hoch in der Luft schweben und Loopings drehen. Unser Flugsimulator macht es möglich – auch ohne Flugschein.

SEITE 40



101 Dämmerung

Welches Ein- und Ausblenden von Bildern in vielen Varianten. Schöne Effekte, die jedem das Arbeiten und Spielen mit dem Computer versüßen.

102 Bitte umblättern...

Einen schnellen Weg zum hintersten Fenster bietet »Shuffle«. Ein pfiffiges Programm, das im Hintergrund läuft und Ihnen einiges an Arbeit abnimmt.

104 Der schnelle Umweg

»Felder« beschleunigt das Abarbeiten von numerischen Feldern erheblich.

Erweiterung

106 Zaubern Sie mit

40 Zaubersprüche: eine Bibliothek voller Funktionen für alle Programmierer – nicht nur für Basic.

Anwendung

121 Meister der Tasten

Wollen Sie schneller auf der Amiga-Tastatur tippen? Mit »Keyboard-Master« erlernen Sie auf faszinierende Weise das 10-Finger-System.

124 Sounds im Eigenbau

Ob Kanonendonner oder akustische Hinweise – der komfortable »Sound-Effekt-Editor« unterstützt Sie beim Erzeugen aller Geräusche für Ihre Programme.

130 Etiketten mit Pfiff

Dieses Programm bringt den Inhalt Ihrer Disketten übersichtlich auf die Etiketten.

136 und läuft und läuft und läuft...

Laufschriften – eine werbewirksame Methode, Informationen auszugeben. »hscroll« generiert sie.

140 Eine universelle Dateiverwaltung

»Uni-Datei« ist schnell und besticht durch sehr komfortable Benutzerführung.

Sonstiges

3 Editorial

4 Inhalt

35 Impressum

143 Checksummer

146 Vorschau

Alle Programme aus Artikeln mit einem Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft.


Solche Bilder können Sie mit dem »IFF-Filter« in den verschiedensten Varianten modifizieren. Wie – das erfahren Sie auf

SEITE 21



Jetzt kommt Ordnung in Ihre Karteien. Die Dateiverwaltung »Uni-Datei« ist derart benutzerfreundlich und schnell, daß keiner mehr darauf verzichten will.

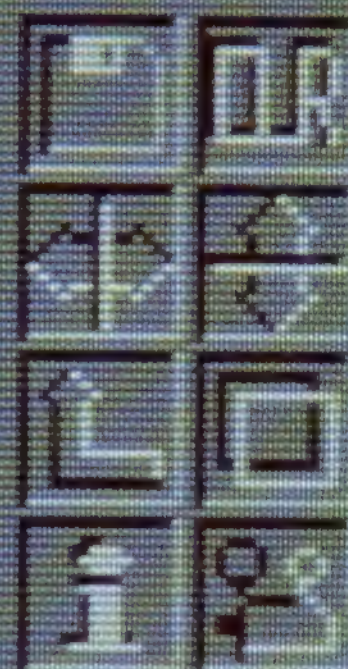
SEITE 140



Sprite-Editor

Sprites gehören zu Spielen wie der Monitor zum Computer. Leider müssen diese auch auf dem Amiga in mühevoller Kleinarbeit erstellt werden. Unser Sprite-Editor beseitigt dieses Problem und macht selbst das Entwickeln von überdimensionalen Sprites zum Kinderspiel.

GRAFIK



Objekt Nr.

661 B

R G B

7

Es gibt kaum Spiele, die ohne Sprites auskommen. Dabei beschränkt sich die Verwendung von bewegten Objekten keineswegs nur auf Actionspiele. Auch Brettspiele, Knobelspiele, Adventures etc. profitieren ungemein durch den Einsatz beweglicher Objekte. Mit der mausgesteuerten Benutzeroberfläche des »Amiga-Object-Editor« (Bild 1) lassen sich Sprites und Animationssequenzen schnell und mühelos erstellen und editieren. Auch DPaint-Brushes lassen sich damit bearbeiten.

Geben Sie den DATA-Lader »AOE_Gen« (Listing 1) mit dem Checksummer ein. Da das Programm einen relativ hohen Speicherbedarf hat, muß vor der Eingabe des Listings im Direktmodus des Basic-Editors »CLEAR, 150000« eingegeben werden. Dies hat auch dann zu geschehen, wenn »AOE_Gen« erneut in den Speicher geladen wird. Nach dem Start des Laders generiert dieser das lauffähige Programm »AOE_V1.2« auf Diskette.

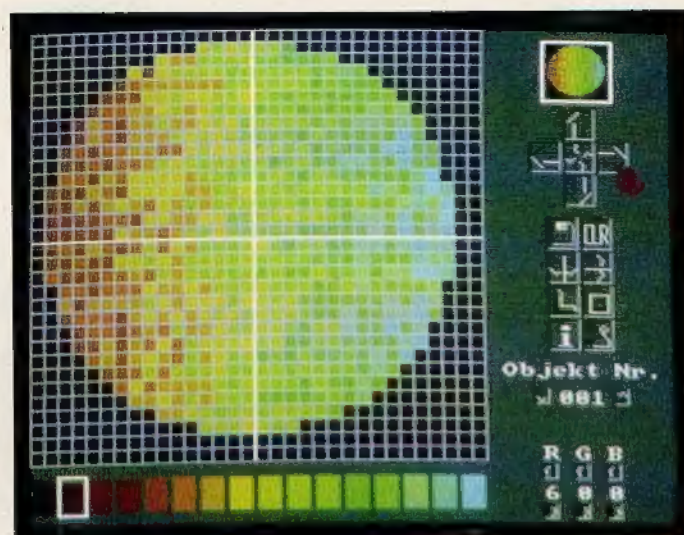


Bild 1. Die Benutzeroberfläche des »Amiga-Object-Editors« mit den verschiedenen Icons

Nach diesem Vorgang, der etwas Zeit beansprucht, kann der Object-Editor gestartet werden. Dabei ist zu beachten, daß das vom Lader generierte File nur vom CLI aus gestartet werden kann, obwohl ein Icon vorhanden wäre.

Sie werden nach dem Start aufgefordert, die Anzahl der Objekte, die bearbeitet werden sollen, einzugeben. Klicken Sie dazu das Eingabefeld an und tragen die gewünschte Objektanzahl ein.

Das Zeichnen eines Objektes ist durch die Maussteuerung sehr einfach. Es stehen 16 verschiedene Farben zur Verfügung, die durch einfaches Anklicken ausgewählt werden. Die Rot-Grün- und Blauanteile der gewählten Farbe können durch die Schalter am unteren rechten Bildschirmrand verändert werden. Gezeichnet wird ein Sprite mit Hilfe der Maus, wobei mit der linken Maustaste Punkte gesetzt und mit der rechten gelöschte Sprites bei »Objekt Nr.« ausgewählt.

Im Zeichenfenster wird das Riesensprite, das in Wirklichkeit aus vier Einzelsprites besteht, stark vergrößert dargestellt. Rechts oben sieht man das Sprite in Originalgröße. Unter diesem Fenster befinden sich vier Pfeile. Durch Anklicken dieser Pfeile kann das aktuelle Sprite in alle vier Himmelsrichtungen verschoben

Icons erleichtern die Arbeit

werden. Klicken Sie das Icon in der Mitte der Pfeile an, so ändern diese ihre Richtung und das Sprite wird nun bei der Verwendung der Pfeilsymbole in Richtung seiner Mittelachse gestaucht. Unter den Pfeil-Icons befinden sich acht weitere Icons, auf deren Funktionen nun eingegangen werden soll: **Icon 1 (Disketten-Icon):** Hier haben Sie die Möglichkeit, Sprites zu laden und zu speichern. Zum Laden muß die entsprechende Option gewählt und die Nummer, unter der das

Adresse	Wert	Bemerkung
\$00000000	"AOE ".L	Kennung der Objektdatei
\$00000004	0000.W	Anzahl der Objekte in der Datei
\$00000008	0000.W	Nun folgen 16 Wörter mit der Farbinformation
\$00000026		Es folgen die Objektdaten: Anzahl der Objekte mal 640 Byte.
Die Objektdaten haben folgenden Aufbau:		
1. Objekt:		1. Bitplane (128 Byte) 4. Bitplane (128 Byte) 5. Maske (128 Byte)
2. Objekt:		1. Bitplane (128 Byte)
Jedem gesetzten Bit in der Maske entspricht ein gesetzter Punkt im Sprite		

Tabelle 1. Der Aufbau eines Files im »AOE«-Format

Objekt im Speicher stehen soll, sowie der Filename eingegeben werden.

Das Programm erkennt, ob es sich um ein »AOE«-Objekt (Amiga-Object-Editor-Objekt), oder um einen DPaint-Brush handelt. DPaint-Brushes müssen in der Größe von 32 x 32 Punkten mit 16 Farben vorlie-

Kopieren und Animieren

gen. Wollen Sie selbsterstellte Sprites auf Diskette ablegen, so wählen Sie »SPEICHERN«. Hier sind die Nummern des ersten und des letzten zu speichernden Objekts angegeben. Es können also beliebige viele

aufeinanderfolgende Sprites in einem File gespeichert werden. Wollen Sie nur ein Sprite speichern, geben Sie einfach zwei gleiche Werte an. Nach Eingabe des Dateinamens werden die Objekte im »AOE«-Format gespeichert. Den Aufbau eines »AEO«-Files sehen Sie in Tabelle 1.

Icon 2 (CLR): Wenn Sie dieses Icon anwählen, wird das aktuelle Sprite gelöscht.

Icon 3 (zwei waagerechte Pfeile): Horizontales Spiegeln des aktuellen Sprites.

Icon 4 (zwei senkrechte Pfeile): Vertikales Spiegeln des aktuellen Sprites.

Icon 5 (geknickter Pfeil): Dre-

hen des aktuellen Sprites um 90 Grad.

Icon 6 (zwei sich überlappende Quadrate): Kopieren von Sprites. Neben der Quell- und Zielnummer des Objekts kann beim Kopieren von mehreren aufeinanderfolgenden Sprites auch die gewünschte Anzahl eingegeben werden.

Icon 7 (»I«): Info zum Programm.

Icon 8 (Bildprojektor-Symbol): Animation von mehreren Sprites. Geben Sie die Nummern des ersten und letzten Sprites der Animationsfolge ein und klicken Sie »FERTIG« an. Daraufhin beginnt die Animation, deren Geschwindig-

keit mit der Maus verändert werden kann. Die Zahl zwischen den Pfeilen gibt die Geschwindigkeit in 1/50-Sekunden an.

Haben Sie Ihre Arbeit beendet und wollen den Editor verlassen, bewegen Sie die Maus in die rechte obere Bildschirm-ecke und drücken den rechten Mausknopf.

Nun steht dem munteren Erstellen von Sprites und Animationssequenzen nichts mehr im Wege. Grenzen für neue Objekte setzt nur die eigene Kreativität — und davon haben Amiga-Besitzer, wie wir wissen, eine ganze Menge.

Michael Bertsch/C. Ewald/
M. Jobst/kn

Programmname: AOE_Gen

Computer: Amiga 500, 2000 mit Kickstart 1.2, 1.3

Sprache: Amiga-Basic 1.2

So geben Sie das Listing ein

Der Objekt-Editor ist in Maschinensprache (Assembler) geschrieben. Um jedem Amiga-Besitzer die Eingabe des Listings zu ermöglichen, haben wir das Programm für Sie umgewandelt. Die abgedruckte Version ist mit Amiga-Basic einzugeben. Diesen Interpreter finden Sie auf der Extras-Diskette, die bei der Auslieferung jedem Amiga beiliegt. Das Programm erzeugt beim Start das lauffähige Programm mit Namen »AOE_V1.2« auf der eingelegten Diskette. Bitte geben Sie das Programm unbedingt mit Checksumme 42 (Seite 143) ein. Die Hinweise zum Checksummer sollten Sie unbedingt beachten.

Listing 1. Der Quellcode des Object-Editors als DATA-Lader. Mit dem Objekt-Editor erzeugen Sie Sprites und Animationssequenzen. Bitte mit dem Checksummer auf Seite 143 eingeben.

```

1 t10 REM Generiert Programm AOE_V1.2
2 ag CLS
3 g0 OPEN "AOE_V1.2" FOR OUTPUT AS 1
4 BS READ ans
5 oa FOR i=1 TO ans
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 FD IF wert1>64 THEN wert1=wert1-55 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7o wert2=ASC(RIGHT$(h$,1))
11 ad IF wert2>64 THEN wert2=wert2-55 ELSE wert2=wert2-48
12 P1 wert=wert1+wert2
13 90 PRINT #1,CHR$(wert);
14 J00 NEXT
15 3n CLOSE 1
16 Ov END
17 R4 DATA 19072
18 1e DATA 00,00,03,F3,00,00,00,00,00,00,00,00,01
19 aJ DATA 00,02,00,00,00,00,00,00,00,00,01
20 UF DATA 00,00,0F,1A,00,00,01,00,00,00
21 gb DATA 03,E9,00,00,0F,1A,4E,F9,00,00
22 72 DATA 2B,94,2C,78,00,04,43,F9,00,00
23 ut DATA 3B,30,42,80,4E,AE,FD,D8,23,00
24 Sy DATA 00,00,3A,3E,43,F9,00,00,3B,41
25 az DATA 42,80,4E,AE,FD,D8,23,00,00,00
26 6d DATA 3A,42,43,F9,00,00,3B,4D,42,80
27 6p DATA 4E,AE,FD,D8,23,00,00,00,3A,46
28 N4 DATA 4E,75,2C,78,00,04,22,79,00,00
29 D7 DATA 3A,3E,4E,AE,FE,62,22,79,00,00
30 T6 DATA 3A,42,4E,AE,FE,62,22,79,00,00
31 bf DATA 3A,46,4E,AE,FE,62,4E,75,2C,79
32 KH DATA 00,00,3A,46,4E,AE,FF,B2,41,F9
33 g4 DATA 00,00,2D,46,4E,AE,FF,3A,23,00
34 9F DATA 00,00,2D,84,23,00,00,00,2D,84
35 at DATA 23,00,00,00,2D,E4,41,F9,00,00
36 PL DATA 2D,66,4E,AE,FF,34,23,00,00,00
37 fb DATA 3A,52,20,40,4E,AE,FE,D4,23,00
38 eK DATA 00,00,3A,62,20,79,00,00,3A,52
39 W3 DATA 23,E8,00,32,00,00,3A,66,23,58
40 BS DATA 00,32,00,00,3A,6E,20,68,00,32
41 ge DATA 23,E8,00,04,00,00,3A,5A,2C,79
42 41 DATA 00,00,3A,3E,20,79,00,00,3A,62
43 v1 DATA 43,F9,00,00,32,6E,30,3C,00,20
44 2u DATA 4E,AE,FF,40,2C,79,00,00,3A,46
45 oE DATA 4E,AE,FE,80,4E,75,2C,79,00,00
46 KJ DATA 3A,46,20,79,00,00,3A,52,4E,AE
47 QW DATA FF,B8,20,79,00,00,2D,84,4E,AE
48 63 DATA FF,BE,4E,AE,FF,2E,4E,75,2C,78
49 2y DATA 00,04,93,C9,4E,AE,FE,DA,28,40
50 3Z DATA 4A,AC,00,AC,66,00,00,18,41,EC
51 pR DATA 00,5C,4E,AE,FE,80,41,EC,00,5C
52 b0 DATA 4E,AE,FE,8C,23,00,00,00,3A,82
53 a0 DATA 4E,75,4A,B9,00,00,3A,82,67,00
54 A1 DATA 00,0C,22,79,00,00,3A,82,4E,AE
55 BD DATA FE,86,42,80,4E,75,2C,79,00,00
56 bd DATA 3A,3E,43,F9,00,00,3A,98,4E,AE
57 71 DATA FF,3A,23,FC,00,00,3B,08,00,00
58 C1 DATA 3A,9C,22,79,00,00,3A,66,30,3C
59 fe DATA 00,10,4E,AE,FE,AA,22,79,00,00
60 6J DATA 3A,66,42,80,42,81,34,3C,00,E0
61 Wd DATA 36,3C,00,E0,4E,AE,FE,CE,22,79
62 dD DATA 00,00,3A,66,30,3C,00,02,4E,AE
63 UQ DATA FE,AA,22,79,00,00,3A,66,42,80
64 RH DATA 4E,AE,FE,A4,22,79,00,00,3A,66
65 JH DATA 10,3C,00,00,4E,AE,FE,9E,42,86
66 KI DATA 22,79,00,00,3A,66,33,46,00,24
67 nV DATA 33,7C,00,00,00,26,30,06,32,3C
68 xL DATA 00,E0,4E,AE,FF,0A,22,79,00,00
69 DZ DATA 3A,66,33,7C,00,00,00,24,33,46
70 HO DATA 00,26,30,3C,00,E0,32,06,4E,AE
71 AZ DATA FF,0A,DC,7C,00,07,BC,7C,00,E7
72 9q DATA 66,C2,22,79,00,00,3A,66,30,3C
73 CF DATA 00,04,4E,AE,FE,AA,22,79,00,00
74 lV DATA 3A,66,33,7C,00,70,00,24,33,7C
75 KQ DATA 00,01,00,26,30,3C,00,70,32,3C
76 85 DATA 00,DF,4E,AE,FF,0A,22,79,00,00
77 dQ DATA 3A,66,33,7C,00,01,00,24,33,7C
78 tV DATA 00,70,00,26,30,3C,00,DF,32,3C
79 OA DATA 00,70,4E,AE,FF,0A,22,79,00,00
80 BR DATA 3A,66,30,3C,00,01,4E,AE,FE,AA
81 oU DATA 22,79,00,00,3A,66,30,3C,00,E1
82 E1 DATA 42,81,34,3C,01,3F,36,3C,00,FF
83 UT DATA 4E,AE,FE,C8,22,79,00,00,3A,66
84 GH DATA 30,3C,00,00,32,3C,00,E1,34,3C
85 d1 DATA 00,E1,36,3C,00,FF,4E,AE,FE,CE
86 oc DATA 22,79,00,00,3A,66,30,3C,00,02
87 EH DATA 4E,AE,FE,AA,22,79,00,00,3A,66
88 31 DATA 30,3C,00,01,4E,AE,FE,A4,30,3C
89 nL DATA 00,02,3B,3C,00,FF,3A,3C,00,07
90 H1 DATA 3C,3C,01,20,3E,3C,00,2E,4E,B9
91 VC DATA 00,00,04,88,30,3C,00,04,38,3C
92 QS DATA 00,FE,3A,3C,00,06,3C,3C,01,21
93 R5 DATA 3E,3C,00,29,4E,B9,00,00,04,88
94 qd DATA 22,79,00,00,3A,66,30,3C,00,10
95 NP DATA 4E,AE,FE,AA,22,79,00,00,3A,66
96 p1 DATA 30,3C,01,00,32,3C,00,08,34,3C
97 w5 DATA 01,1F,36,3C,00,27,4E,AE,FE,CE
98 fv DATA 4E,F9,00,00,32,AE,3C,3C,00,10
99 tP DATA 30,06,22,79,00,00,3A,66,4E,AE
100 Jx DATA FE,AA,30,1D,32,1D,34,1D,36,1D
101 Xn DATA 22,79,00,00,3A,66,4E,AE,FE,CE
102 II DATA DC,7C,00,01,8C,7C,00,20,66,D8
103 kE DATA 30,3C,00,02,38,3C,00,0F,3A,3C
104 lV DATA 00,E7,3C,3C,00,1B,3E,3C,00,F9
105 4E DATA 4E,B9,00,00,04,88,30,3C,00,04
106 Zk DATA 3B,3C,00,0E,3A,3C,00,E6,3C,3C
107 01 DATA 00,1C,3E,3C,00,FA,4E,B9,00,00
108 Uh DATA 04,88,2C,79,00,00,3A,46,20,79
109 WR DATA 00,00,3A,66,43,F9,00,00,34,36
110 mH DATA 30,3C,01,09,32,3C,00,50,4E,AE
111 Tf DATA FF,8E,43,F9,00,00,34,FE,20,79
112 D2 DATA 00,00,3A,66,30,3C,01,00,32,3C
113 Ue DATA 00,66,4E,AE,FF,8E,43,F9,00,00
114 as DATA 35,9E,20,79,00,00,3A,66,30,3C
115 tH DATA 00,F8,32,3C,00,BC,4E,AE,FF,8E
116 NL DATA 43,F9,00,00,35,C6,20,79,00,00
117 pZ DATA 3A,66,30,3C,00,FB,32,3C,00,E2
118 D1 DATA 4E,AE,FF,8E,4B,F9,00,00,3B,5F
119 Rm DATA 3A,3C,00,0A,3C,3C,00,E8,3E,3C
120 QW DATA 00,B4,4E,B9,00,00,04,18,4B,F9
121 Ts DATA 00,00,3B,74,3A,3C,00,03,3C,3C
122 EI DATA 01,03,3E,3C,00,C2,4E,B9,00,00
123 Eh DATA 04,18,4B,F9,00,00,3B,69,3A,3C
124 xJ DATA 00,05,3C,3C,00,FB,3E,3C,00,DE
125 21 DATA 4E,B9,00,00,04,18,4E,B9,00,00

```


126 R3 DATA 05,C2,4E,75,2C,79,00,00,3A,3E
 127 z1 DATA 22,79,00,00,3A,6E,33,46,00,24
 128 p3 DATA 33,47,00,26,30,3C,00,01,4E,AE
 129 G8 DATA FE,9E,22,79,00,00,3A,6E,42,80
 130 BQ DATA 4E,AE,FE,AA,22,79,00,00,3A,6E
 131 mS DATA 20,4D,30,05,4E,AE,FF,04,22,79
 132 OK DATA 00,00,3A,6E,DC,7C,00,01,DE,7C
 133 Db DATA 00,01,33,46,00,24,33,47,00,26
 134 b4 DATA 42,80,4E,AE,FE,9E,22,79,00,00
 135 cw DATA 3A,6E,30,3C,00,03,4E,AE,FE,AA
 136 up DATA 22,79,00,00,3A,6E,20,4D,30,05
 137 YL DATA 4E,AE,FF,04,4E,75,2C,79,00,00
 138 07 DATA 3A,3E,22,79,00,00,3A,6E,4E,AE
 139 ga DATA FE,AA,22,79,00,00,3A,6E,33,44
 140 Rn DATA 00,24,33,45,00,26,30,06,32,05
 141 j8 DATA 4E,AE,FF,0A,22,79,00,00,3A,6E
 142 Qv DATA 30,06,32,07,4E,AE,FF,0A,22,79
 143 wJ DATA 00,00,3A,6E,30,04,32,07,4E,AE
 144 5y DATA FF,0A,22,79,00,00,3A,6E,30,04
 145 S1 DATA 32,05,4E,AE,FF,0A,4E,75,42,87
 146 yx DATA 41,F9,00,00,32,AE,30,18,32,18
 147 3p DATA 34,18,36,18,B0,45,62,00,00,1A
 148 13 DATA B2,46,62,00,00,14,34,45,65,00
 149 no DATA 00,0E,B6,46,65,00,00,08,4E,F9
 150 KH DATA 00,00,05,20,DE,7C,00,01,BE,7C
 151 de DATA 00,10,66,D0,08,39,00,06,00,BF
 152 HY DATA E0,01,67,F6,60,00,26,FE,23,F9
 153 C9 DATA 00,00,3A,66,00,00,3A,6E,33,C7
 154 PP DATA 00,00,3A,74,3E,39,00,00,3A,72
 155 1v DATA CE,FC,00,08,28,47,D9,FC,00,00
 156 N8 DATA 32,AE,61,00,00,62,61,00,00,68
 157 wJ DATA 30,3C,00,01,61,00,FF,38,61,00
 158 yz DATA 00,5C,30,3C,00,01,61,00,FF,2C
 159 rd DATA 42,87,3E,39,00,00,3A,74,33,C7
 160 uQ DATA 00,00,3A,72,CE,FC,00,08,28,47
 161 SW DATA D9,FC,00,00,32,AE,61,00,00,2C
 162 AE DATA 61,00,00,32,30,3C,00,02,61,00
 163 Ph DATA FF,02,61,00,00,26,30,3C,00,04
 164 b7 DATA 61,00,FE,F6,61,00,00,2C,08,39
 165 IQ DATA 00,06,00,BF,80,01,67,F6,60,00
 166 BQ DATA 26,78,3E,1C,3A,1C,3C,1C,3E,1C
 167 CH DATA 4E,75,9E,7C,00,01,9A,7C,00,01
 168 pz DATA DC,7C,00,01,DE,7C,00,01,4E,75
 169 TU DATA 42,80,30,39,00,00,3A,72,C0,FC
 170 HY DATA 00,02,11,40,D3,FC,00,00,32,8E
 171 8N DATA 30,11,3E,00,61,00,00,48,13,C0
 172 98 DATA 00,00,3B,72,30,07,EB,48,61,00
 173 wu DATA 00,3A,13,00,00,00,3B,70,30,07
 174 DW DATA E0,48,61,00,00,2C,13,00,00,00
 175 HS DATA 3B,6E,4B,F9,00,00,3B,6E,3A,3C
 176 Vv DATA 00,06,3C,3C,00,00,FB,3E,3C,00,F2
 177 gp DATA 23,F9,00,00,3A,66,00,00,3A,6E
 178 vb DATA 4E,B9,00,00,04,18,4E,75,C0,7C
 179 x3 DATA 00,0F,D0,7C,00,30,B0,7C,00,3A
 180 LO DATA 65,00,00,06,D0,7C,00,07,4E,75
 181 Ao DATA 2C,79,00,00,3A,7E,8A,FC,00,07
 182 5q DATA 8C,FC,00,07,CA,BC,00,00,FF,FF
 183 yR DATA CC,BC,00,00,FF,FF,20,05,22,06
 184 or DATA C0,FC,00,07,C2,FC,00,07,D0,7C
 185 iq DATA 00,01,D2,7C,00,01,3A,00,36,01
 186 04 DATA D4,7C,00,05,D6,7C,00,05,48,E7
 187 jZ DATA F0,00,30,39,00,00,3A,72,D0,7C
 188 jB DATA 00,10,22,79,00,00,3A,66,4E,AE
 189 uN DATA FE,AA,22,79,00,00,3A,66,4C,DF
 190 AG DATA 00,0F,4E,AE,FE,CE,43,F9,00,00
 191 Vv DATA 3A,98,30,39,00,00,3A,72,D0,7C
 192 nK DATA 00,10,4E,AE,FE,AA,43,F9,00,00
 193 Wa DATA 3A,98,30,05,32,06,4E,AE,FE,BC
 194 Bc DATA 22,79,00,00,3A,66,30,05,32,06
 195 EJ DATA D0,7C,01,00,D2,7C,00,08,4E,AE
 196 eK DATA FE,BC,4E,75,42,87,41,F9,00,00
 197 Hd DATA 33,2E,30,18,32,18,34,18,36,18
 198 jF DATA B0,45,62,00,00,1A,34,46,62,00
 199 Ba DATA 00,14,B4,45,65,00,00,0E,B6,46
 200 gY DATA 65,00,00,08,4E,F9,00,00,07,1A
 201 qR DATA DE,7C,00,01,BE,7C,00,16,66,D0
 202 AC DATA 08,39,00,06,00,BF,ED,01,67,F6
 203 qu DATA 60,00,25,04,CE,FC,00,04,20,47
 204 jE DATA D1,FC,00,00,33,DE,22,50,4E,D1
 205 7F DATA 08,39,00,06,00,BF,ED,01,67,F6
 206 9I DATA 23,F9,00,00,3A,66,00,00,3A,6E
 207 hK DATA 60,00,24,DC,0C,79,00,01,00,00
 208 2D DATA 3A,7C,67,00,01,84,20,79,00,00
 209 DM DATA 3B,10,61,00,00,26,20,79,00,00

210 DI DATA 3B,14,61,00,00,1C,20,79,00,00
 211 nL DATA 3B,18,61,00,00,12,20,79,00,00
 212 eY DATA 3B,1C,61,00,00,08,61,00,20,2E
 213 Vd DATA 60,AE,2E,28,00,7C,24,48,22,48
 214 7v DATA D3,FC,00,00,00,80,D1,FC,00,00
 215 nI DATA 00,7C,20,3C,00,00,00,1E,23,20
 216 5d DATA 51,C8,FF,FC,24,87,4E,75,0C,79
 217 bR DATA 00,01,00,00,3A,7C,67,00,01,78
 218 3Q DATA 20,79,00,00,3B,10,61,00,00,28
 219 Fv DATA 20,79,00,00,3B,14,61,00,00,1E
 220 5V DATA 20,79,00,00,3B,18,61,00,00,14
 221 eP DATA 20,79,00,00,3B,1C,61,00,00,0A
 222 hU DATA 61,00,1F,D0,60,00,FF,50,2E,10
 223 93 DATA 24,48,22,48,D1,FC,00,00,00,04
 224 2z DATA 20,3C,00,00,00,1E,22,D8,51,C8
 225 0x DATA FF,FC,25,47,00,7C,4E,75,0C,79
 226 j1 DATA 00,01,00,00,3A,7C,67,00,01,6E
 227 C2 DATA 20,79,00,00,3B,10,61,00,00,28
 228 Q1 DATA 20,79,00,00,3B,14,61,00,00,1E
 229 Ee DATA 20,79,00,00,3B,18,61,00,00,14
 230 1Y DATA 20,79,00,00,3B,1C,61,00,00,0A
 231 s2 DATA 61,00,1F,76,60,00,FE,F6,20,3C
 232 Hb DATA 00,00,00,1F,22,10,E2,99,20,C1
 233 Fe DATA 51,C8,FF,F6,4E,75,0C,79,00,01
 234 9a DATA 00,00,3A,7C,67,00,01,62,20,79
 235 ea DATA 00,00,3B,10,61,00,00,28,20,79
 236 ac DATA 00,00,3B,14,61,00,00,1E,20,79
 237 4c DATA 00,00,3B,18,61,00,00,14,20,79
 238 2t DATA 00,00,3B,1C,61,00,00,0A,61,00
 239 1U DATA 1F,28,60,00,FE,A8,20,3C,00,00
 240 CH DATA 00,1F,22,10,E3,99,20,C1,51,C8
 241 rH DATA FF,F6,4E,75,0C,79,00,01,00,00
 242 x4 DATA 3A,7C,67,00,01,5A,2C,79,00,00
 243 0Q DATA 3A,46,20,79,00,00,3A,66,43,F9
 244 fH DATA 00,00,3A,9A,20,3C,00,00,01,09
 245 wt DATA 22,3C,00,00,00,50,4E,AE,FF,8E
 246 tN DATA 33,FC,00,01,00,00,3A,7C,60,00
 247 fJ DATA FE,5C,20,79,00,00,3B,10,61,00
 248 uQ DATA 00,28,20,79,00,00,3B,14,61,00
 249 MR DATA 00,1E,20,79,00,00,3B,18,61,00
 250 R1 DATA 00,14,20,79,00,00,3B,1C,61,00
 251 En DATA 00,0A,61,00,1E,AC,60,00,FE,2C
 252 1e DATA 24,48,22,48,D3,FC,00,00,00,40
 253 FU DATA D1,FC,00,00,00,44,20,3C,00,00
 254 AM DATA 00,0F,22,D8,51,C8,FF,FC,42,AA
 255 dD DATA 00,7C,4E,75,20,79,00,00,3B,10
 256 si DATA 61,00,00,28,20,79,00,00,3B,14
 257 X1 DATA 61,00,00,1E,20,79,00,00,3B,18
 258 tY DATA 61,00,00,14,20,79,00,00,3B,1C
 259 Tr DATA 61,00,00,0A,61,00,1E,5A,60,00
 260 Bf DATA FD,DA,24,48,22,48,D3,FC,00,00
 261 Bn DATA 00,3C,D1,FC,00,00,00,40,20,3C
 262 zk DATA 00,00,00,0F,21,21,51,C8,FF,FC
 263 VJ DATA 42,92,4E,75,20,79,00,00,3B,10
 264 Oq DATA 61,00,00,28,20,79,00,00,3B,14
 265 fQ DATA 61,00,00,1E,20,79,00,00,3B,18
 266 16 DATA 61,00,00,14,20,79,00,00,3B,1C
 267 Mf DATA 61,00,00,0A,61,00,1E,0A,60,00
 268 50 DATA FD,8A,20,3C,00,00,00,1F,22,10
 269 EJ DATA E3,49,20,C1,51,C8,FF,F6,4E,75
 270 te DATA 20,79,00,00,3B,10,61,00,00,28
 271 5i DATA 20,79,00,00,3B,14,61,00,00,1E
 272 vL DATA 20,79,00,00,3B,18,61,00,00,14
 273 SF DATA 20,79,00,00,3B,1C,61,00,00,0A
 274 p1 DATA 61,00,1D,C8,60,00,FD,48,20,3C
 275 oR DATA 00,00,00,1F,22,10,48,41,E2,49
 276 uL DATA 4B,41,20,C1,51,C8,FF,F4,4E,75
 277 xR DATA 2C,79,00,00,3A,46,20,79,00,00
 278 Me DATA 3A,66,43,F9,00,00,3A,36,20,3C
 279 UP DATA 00,00,01,09,22,3C,00,00,00,50
 280 4w DATA 4E,AE,FF,8E,33,FC,00,00,00,00
 281 Ff DATA 3A,7C,60,00,FD,04,41,F9,00,00
 282 V1 DATA 20,96,20,79,00,00,3A,46,4E,AE
 283 Qt DATA FF,34,23,C0,00,00,3A,56,67,00
 284 kJ DATA FC,EA,20,40,23,EB,00,32,00,00
 285 bU DATA 3A,6A,22,68,00,32,30,3C,00,01
 286 vD DATA 2C,79,00,00,3A,3E,4E,AE,FE,AA
 287 VI DATA 22,79,00,00,3A,6A,30,3C,00,01
 288 wX DATA 4E,AE,FE,AA,22,79,00,00,3A,6A
 289 mR DATA 30,3C,00,00,32,3C,00,00,3A,3C
 290 po DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
 291 3N DATA 23,F9,00,00,3A,6A,00,00,3A,6E
 292 E2 DATA 4B,F9,00,00,3B,77,3A,3C,00,05
 293 BE DATA 3C,3C,00,5C,3E,3C,30,5B,4E,B9

294 4p DATA 00,00,04,18,4B,F9,00,00,3B,7C
 295 a2 DATA 3A,3C,00,09,3C,3C,00,4C,3E,3C
 296 q1 DATA 00,6D,4E,B9,00,00,04,18,4B,F9
 297 1a DATA 00,00,3C,13,3A,3C,00,07,3C,3C
 298 6I DATA 00,54,3E,3C,00,7F,4E,B9,00,00
 299 9D DATA 04,18,30,3C,00,02,38,3C,00,46
 300 10 DATA 3A,3C,00,50,3C,3C,00,9A,3E,3C
 301 B8 DATA 00,60,4E,B9,00,00,04,88,30,3C
 302 rs DATA 00,02,38,3C,00,46,3A,3C,00,62
 303 j6 DATA 3C,3C,00,9A,3E,3C,00,72,4E,B9
 304 oT DATA 00,00,04,88,30,3C,00,02,38,3C
 305 nw DATA 00,46,3A,3C,00,74,3C,3C,00,9A
 306 IT DATA 3E,3C,00,84,4E,B9,00,00,04,88
 307 1t DATA 08,39,00,06,00,BF,ED,01,67,F6
 308 kr DATA 08,39,00,06,00,BF,ED,01,66,F6
 309 rg DATA 22,79,00,00,3A,56,3A,29,00,0E
 310 fs DATA 3C,29,00,0C,BA,7C,00,46,65,00
 311 CX DATA 00,1E,BC,7C,00,50,65,00,00,16
 312 Vw DATA BA,7C,00,9A,62,00,00,0E,BC,7C
 313 D0 DATA 00,60,62,00,00,06,60,00,00,52
 314 ad DATA BA,7C,00,46,65,00,00,1E,BC,7C
 315 Jr DATA 00,62,65,00,00,16,BA,7C,00,9A
 316 1Y DATA 62,00,00,0E,BC,7C,00,72,62,00
 317 eq DATA 00,06,60,00,00,F4,BA,7C,00,46
 318 gJ DATA 65,9A,BC,7C,00,74,65,9A,BA,7C
 319 g3 DATA 00,9A,62,8E,BC,7C,00,84,62,88
 320 eY DATA 2C,79,00,00,3A,46,20,79,00,00
 321 1t DATA 3A,56,4E,AE,FF,8E,60,00,FB,70
 322 ga DATA 2C,79,00,00,3A,46,20,79,00,00
 323 EZ DATA 3A,56,4E,AE,FF,8E,20,79,00,00
 324 04 DATA 3A,62,30,3C,00,1F,32,3C,00,0D
 325 Fm DATA 34,01,36,01,2C,79,00,00,3A,3E
 326 Rr DATA 4E,AE,FE,ED,20,79,00,00,3A,62
 327 14 DATA 30,3C,00,1E,32,3C,00,03,34,01
 328 MR DATA 36,01,4E,AE,FE,ED,20,79,00,00
 329 18 DATA 3A,46,23,FC,00,00,2D,F6,00,00
 330 xo DATA 2D,AB,41,F9,00,00,2D,96,4E,AE
 331 qW DATA FF,34,23,C0,00,00,3A,56,20,40
 332 xF DATA 23,EB,00,32,00,00,3A,6A,23,EB
 333 6F DATA 00,32,00,00,3A,6E,22,68,00,32
 334 er DATA 30,3C,00,01,20,79,00,00,3A,3E
 335 7L DATA 4E,AE,FE,AA,22,79,00,00,3A,6A
 336 T8 DATA 30,3C,00,01,4E,AE,FE,AA,22,79
 337 6A DATA 00,00,3A,6A,30,3C,00,00,32,30
 338 Jd DATA 00,00,3A,3C,00,00,DF,36,3C,00,54
 339 Vt DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
 340 kh DATA 30,3C,00,00,32,30,00,55,34,3C
 341 m5 DATA 00,5D,36,3C,00,61,4E,AE,FE,CE
 342 uD DATA 22,79,00,00,3A,6A,30,3C,00,81
 343 au DATA 32,3C,00,55,34,3C,00,DF,36,3C
 344 1S DATA 00,61,4E,AE,FE,CE,22,79,00,00
 345 w4 DATA 3A,6A,30,3C,00,00,32,3C,00,61
 346 eD DATA 34,3C,00,DF,36,3C,00,72,4E,AE
 347 NW DATA FE,CE,22,79,00,00,3A,6A,30,3C
 348 eT DATA 00,00,32,3C,00,73,34,3C,00,24
 349 VT DATA 36,3C,00,7F,4E,AE,FE,CE,22,79
 350 hB DATA 00,00,3A,6A,30,3C,00,B8,32,30
 351 xW DATA 00,73,34,3C,00,DF,36,3C,00,7F
 352 16 DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
 353 9F DATA 30,3C,00,00,32,3C,00,7F,34,3C
 354 rQ DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
 355 vC DATA 4B,F9,00,00,3B,85,3A,3C,00,11
 356 jP DATA 3C,3C,00,27,3E,3C,00,4F,4E,B9
 357 AK DATA 00,00,04,18,4B,F9,00,00,3B,96
 358 3x DATA 3A,3C,00,09,3C,3C,00,27,3E,3C
 359 ff DATA 00,6B,4E,B9,00,00,04,18,4B,F9
 360 In DATA 00,00,3B,9F,3A,3C,00,06,3C,3C
 361 YB DATA 00,58,3E,3C,00,91,4E,B9,00,00
 362 47 DATA 04,18,30,3C,00,02,38,3C,00,54
 363 uY DATA 3A,3C,00,87,3C,3C,00,8C,3E,3C
 364 GF DATA 00,96,4E,B9,00,00,04,88,30,39
 365 39 DATA 00,00,2E,82,08,00,00,07,66,00
 366 Wv DATA 00,36,08,39,00,06,00,BF,ED,01
 367 Hh DATA 66,EB,22,79,00,00,3A,56,3A,29
 368 30 DATA 00,0E,3C,29,00,0C,BA,7C,00,54
 369 +2 DATA 65,D4,BC,7C,00,87,65,CE,BA,7C
 370 P6 DATA 00,BC,62,CB,BC,7C,00,96,62,C2
 371 BR DATA 60,00,00,0E,30,39,00,00,2E,82
 372 gD DATA 08,00,00,07,66,F4,0C,B9,00,00
 373 jV DATA 00,00,00,00,2E,62,66,00,00,00

Listing 1. (Fortsetzung)

SuperbaseTM FÜR JEDEN GELDBEUTEL

ca. **DM 89,-***
(sFr. 61,90,- / 6S 757,-*)



Amiga Superbase (Bookware)

Wegen seiner Bedienerfreundlichkeit die optimale Einstiegs-Datenbank. Per Maus/Click erstellen Sie Ihre Datenmaske. Aufnahme, Auswahl und Ausgabe der Daten erfolgen über ein Bedienerfeld, das dem eines Videorecorders sehr ähnlich ist. Selbstverständlich können Sie Amiga-Gratiken ebenso verwerten und anzeigen lassen wie Zahlen und Texte. Briefmarkensammler zum Beispiel könnten somit die digitalisierten Abbilder ihrer Schätze mit den zugehörigen Daten auf den Monitor bringen. Das System ist relational, Sie können also Einträge verschiedener Datenbanken miteinander verknüpfen. Die Daten lassen sich als Formular oder Liste auf dem Bildschirm oder Drucker ausgeben. Vorbestellbar also die Zeit des Chaos in Platten-, Dia- oder anderen Sortimenten!
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte Arbeitsspeicher
1989, 188 Seiten, inkl. Programmdiskette
ISBN 3 89090 791-1 Bestell-Nr. 90791

DM 199,-*
(sFr. 179,-* 6S 1990,-*)



Superbase 2 Amiga (deutsch)

Wegen seiner Verkaufszahlen der Nummer unter den Datenbanken für den Hausgebrauch ebenso geeignet wie für das Büro. Daten und Bilder lassen sich mit diesem mächtigen relationalen Datenbanksystem fast spielerisch verwalten. Eine neue Form der Lagerhaltung ist somit beispielsweise möglich. Nicht mehr nur die Daten, sondern auch ein digitalisiertes Bild jedes Artikels erscheinen auf dem Monitor. Bis zu 16 Millionen Datensätze pro Datei und eine unbegrenzte Anzahl geöffneter Dateien sind erlaubt. Die Ein- oder Ausgabe erfolgt in Listen oder Formularen, die Sie sich am Bildschirm erstellen können. Weitere Leistungsmerkmale: Textverarbeitungsprogramm integriert, Serienbrieffunktion, speicherbare Masken, Etikettendruck u.v.m.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mind. 512 Kbyte RAM (empfohlen 1 Mbyte)
Bestell-Nr. 54110
Upgrade von Superbase 2 auf Superbase Professional, 51672U, DM 199,-* (sFr. 179,-*/6S 1990,-*)

DM 399,-*
(sFr. 359,-* 6S 3890,-*)



Superbase Professional

Neben den Funktionen von Superbase 2 bietet Ihnen diese Professional Version zwei erhebliche Vorteile: zum einen den Formular-Editor, mit dem Sie sich grafische Ein- und Ausgabemasken in grandioser Qualität schaffen können, und zum anderen beinhaltet sie eine Programmiermöglichkeit. Die Programmiersprache DML ist zwar einfach erlernbar wie Basic, unterstützt die Möglichkeiten von Superbase jedoch vollständig. Somit sind der Flexibilität der Datenbank keine Grenzen gesetzt, sei es bei der individuellen Bedienung, sei es bei der Möglichkeit Daten zu selektieren, darzustellen und auf dem Drucker auszugeben.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte RAM (empfohlen 1 Mbyte RAM, Festplatte) Bestell-Nr. 51672
Upgrade Superbase Professional auf Superbase Professional Entwickler-Paket Bestell-Nr. 54136U, gegen Einsendung der Originaldiskette DM 199,-* (sFr. 179,-*/6S 1990,-*)

DM 599,-*
(sFr. 539,-* 6S 5890,-*)



Superbase Professional Entwickler-Paket

Superbase Professional Entwickler-Paket ist die konsequente Weiterentwicklung der herausragenden Datenbank Superbase Professional. Im Anwender-Paket wurden die Tastaturbedienung, die Einbaumöglichkeit von Hilfefenstern, die Import-/Export-Funktion, der Formulareditor u.v.m. erheblich erweitert. dBase, Lotus u.a. Dateien können nun komfortabel eingelesen werden. Ein leistungsstarkes DFU-Programm wurde integriert, um die Datenfernübertragung zu erleichtern. Für die Programmierer stehen nun weitere Befehle zur Verfügung, die zum Beispiel Transaktionen von Daten möglich machen. Das mitgelieferte Run-time-Modul ermöglicht es Entwicklern, ihre Anwendungen auch denen zugänglich zu machen, die das Superbase-Professional-Entwickler-Paket nicht besitzen.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte RAM (empfohlen 1 Mbyte RAM, Festplatte) Lieferbar 2 Quartal 1989, Bestell-Nr. 54136

*Unverbindliche Preisempfehlung



Markt & Technik-Bücher und Software erhalten Sie bei Ihrem Buchhändler, in Computerelektrogeschäften und in den Fachabteilungen der Warenhäuser.

WWW.HOMECOMPUTERWORLD.COM

374 hq DATA 23,FC,00,00,00,01,00,00,2E,62
375 xr DATA 2C,79,00,00,00,3A,46,20,79,00,00
376 ki DATA 3A,56,4E,AE,FF,B8,23,FC,00,00,00
377 gl DATA 00,00,00,00,2D,88,2C,79,00,00,00
378 lf DATA 3A,3E,20,79,00,00,3A,62,43,F9
379 oh DATA 00,00,32,6E,30,3C,00,2D,4E,AE
380 mh DATA FF,40,2C,79,00,00,3A,42,24,3C
381 vr DATA 00,00,03,ED,22,3C,00,00,2E,EA
382 7m DATA 4E,AE,FF,E2,23,C0,00,00,3A,86
383 N1 DATA 67,00,01,06,22,00,26,3C,00,00,00
384 hb DATA 00,04,24,3C,00,00,00,00,4E,AE
385 CJ DATA FF,D6,67,00,00,CC,0C,B9,41,4F
386 jK DATA 45,20,00,00,00,00,66,00,00,00,00
387 xS DATA 22,39,00,00,00,3A,86,26,3C,00,00
388 pv DATA 00,02,24,3C,00,00,00,00,4E,AE
389 qI DATA FF,D6,67,00,00,00,44,20,39,00,00
390 ep DATA 2E,62,D0,79,00,00,00,00,40,B0,B9
391 xu DATA 00,00,3A,78,62,00,00,00,8E,22,39
392 Iy DATA 00,00,3A,86,26,3C,00,00,00,00,20
393 Sr DATA 24,3C,00,00,32,6E,4E,AE,FF,D6
394 k3 DATA 67,00,00,00,44,22,62,24,39,00,00
395 Ad DATA 2E,62,94,BC,00,00,00,00,01,C4,FC
396 mo DATA 02,80,D4,B9,00,00,00,3A,44,22,83
397 51 DATA 36,39,00,00,00,00,04,C6,FC,02,80
398 xy DATA 22,39,00,00,00,3A,86,4E,AE,FF,D6
399 iX DATA 67,00,00,42,4E,B9,00,00,00,2A,F2
400 dl DATA 2C,79,00,00,00,3A,3E,20,79,00,00
401 j8 DATA 3A,62,43,F9,00,00,32,6E,30,3C
402 eV DATA 00,20,4E,AE,FF,40,4E,B9,00,00
403 x5 DATA 27,A6,4E,B9,00,00,00,05,C2,2C,79
404 an DATA 00,00,3A,42,22,39,00,00,3A,86
405 ei DATA 4E,AE,FF,DC,60,00,00,00,00,20,79
406 14 DATA 00,00,2D,B4,2C,79,00,00,00,3A,46
407 pn DATA 4E,AE,FF,A0,2C,79,00,00,00,3A,42
408 A0 DATA 22,39,00,00,00,3A,86,4E,AE,FF,DC
409 gS DATA 60,00,00,00,20,79,00,00,00,2D,B4
410 Ud DATA 2C,79,00,00,00,3A,46,4E,AE,FF,A0
411 10 DATA 60,00,00,00,00,00,00,00,00,00,00
412 6F DATA 00,00,00,00,00,66,BC,22,39,00,00
413 VQ DATA 3A,86,26,3C,00,00,00,00,04,24,3C
414 gh DATA 00,00,00,00,4E,AE,FF,D6,67,A4
415 gb DATA 0C,B9,00,00,00,00,04,00,00,00,00
416 Bk DATA 64,98,22,39,00,00,00,3A,86,26,3C
417 SE DATA 00,00,04,00,24,3C,00,00,00,00,00
418 04 DATA 4E,AE,FF,D6,41,39,00,00,00,00,00
419 iY DATA 20,18,B0,BC,49,4C,42,4D,66,00
420 eu DATA FF,72,20,18,B0,BC,42,4D,48,44
421 2u DATA 67,00,00,00,3E,B1,FC,00,00,04,00
422 Sn DATA 66,EC,60,00,00,00,32,20,18,B0,BC
423 1j DATA 43,4D,41,50,67,00,00,00,57,B1,FC
424 j8 DATA 00,00,04,00,66,EC,60,00,00,FF,42
425 xS DATA 20,18,B0,BC,42,4F,44,57,67,00
426 jH DATA 00,6A,B1,FC,00,00,04,00,66,EC
427 du DATA 60,00,FF,2A,D1,FC,00,00,00,04
428 ze DATA 20,18,B0,BC,00,20,00,20,66,00
429 0E DATA FF,18,D1,FC,00,00,00,04,10,18
430 w1 DATA 80,3C,00,04,66,00,FF,08,D1,FC
431 My DATA 00,00,00,00,00,60,A6,D1,FC,00,00
432 g4 DATA 00,04,43,F9,00,00,00,3A,86,20,3C
433 0w DATA 00,00,00,0F,42,41,42,42,12,18
434 34 DATA 89,49,84,41,42,41,12,18,84,41
435 n4 DATA 42,41,12,18,84,49,84,41,32,C2
436 1m DATA 51,C8,FF,E4,60,51,C8,FF,E4,FC,00,00
437 16 DATA 00,04,20,39,00,00,2E,62,90,3C
438 Mk DATA 00,00,00,01,00,FC,02,80,D0,B9
439 4f DATA 00,00,00,3A,4A,22,40,24,40,26,40
440 Ph DATA 28,40,D5,FC,00,00,00,80,D7,FC
441 Nv DATA 00,00,01,00,D9,FC,00,00,01,80
442 Tn DATA 20,3C,00,00,00,1F,22,D8,24,D8
443 2i DATA 26,D8,28,D8,51,C8,FF,E4,FC,00,00
444 Xn DATA FE,42,2C,79,00,00,3A,46,20,79
445 H8 DATA 00,00,3A,56,4E,AE,FF,B8,20,79
446 q2 DATA 00,00,3A,62,30,3C,00,00,1F,32,3C
447 EY DATA 00,0D,3A,01,36,01,2C,79,00,00
448 AP DATA 3A,3E,4E,AE,FE,E0,20,79,00,00
449 xN DATA 3A,62,30,3C,00,1E,32,3C,00,03
450 Zn DATA 34,01,36,01,4E,AE,FE,E0,2C,79
451 tF DATA 00,00,3A,46,23,FC,00,00,2F,3E
452 yd DATA 00,00,2D,A8,41,F9,00,00,2D,96
453 0C DATA 4E,AE,FF,34,23,C0,00,00,3A,56
454 5p DATA 20,40,23,88,00,32,00,00,3A,6A
455 0i DATA 23,88,00,32,00,00,3A,6E,22,68
456 CE DATA 0D,32,30,3C,00,01,2C,79,00,00
457 0W DATA 3A,3E,4E,AE,FE,A8,22,79,00,00

458 a DATA 3A,6A,30,3C,00,01,4E,AE,FE,A4
459 72 DATA 22,79,00,00,3A,6A,30,3C,00,00
460 30 DATA 32,3C,00,00,34,3C,00,0F,36,3C
461 7C DATA 00,00,4E,AE,FE,CE,22,79,00,00
462 3C DATA 3A,6A,30,3C,00,00,32,3C,00,55
463 7E DATA 34,3C,00,00,36,3C,00,01,4E,AE
464 2P DATA 00,00,CE,22,79,00,00,3A,6A,30,3C
465 4E DATA 00,59,32,3C,00,55,3A,3C,00,85
466 4C DATA 36,3C,00,61,4E,AE,FE,CE,22,79
467 45 DATA 00,00,3A,6A,30,3C,00,00,32,3C
468 7A DATA 00,55,34,3C,00,0F,36,3C,00,61
469 4C DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
470 4E DATA 30,3C,00,00,32,3C,00,61,34,3C
471 X0 DATA 00,0F,36,3C,00,00,72,4E,AE,FE,CE
472 FM DATA 22,79,00,00,3A,6A,30,3C,00,00
473 4A DATA 32,3C,00,73,34,3C,00,1D,36,3C
474 7E DATA 00,7F,4E,AE,FE,CE,22,79,00,00
475 2A DATA 3A,6A,30,3C,00,00,51,32,3C,00,73
476 4V DATA 34,3C,00,0F,36,3C,00,7F,4E,AE
477 7C DATA 7E,CE,22,79,00,00,3A,6A,30,3C
478 F2 DATA 00,00,32,3C,00,00,7F,34,3C,00,DF
479 12 DATA 36,3C,00,0F,4E,AE,FE,CE,4B,F9
480 Nb DATA 00,00,3B,A5,3A,3C,00,14,3C,3C
481 4A DATA 00,1F,3E,3C,00,00,4E,B9,00,00
482 2G DATA 04,1B,4B,F9,00,00,3B,B9,3A,3C
483 40 DATA 00,04,3C,3C,00,00,5F,3E,3C,00,5D
484 4S DATA 4E,B9,00,00,04,18,4B,F9,00,00
485 1S DATA 3B,96,3A,3C,00,00,09,3C,00,1F
486 0A DATA 3E,3C,00,6B,4E,B9,00,00,04,18
487 78 DATA 4B,F9,00,00,3B,9F,3A,3C,00,06
488 4C DATA 3C,3C,00,00,3E,3C,00,91,4E,B9
489 72 DATA 00,00,04,18,30,3C,00,02,38,3C
490 4C DATA 00,54,3A,3C,00,87,3C,3C,00,8C
491 4A DATA 3E,3C,00,96,4E,B9,00,00,04,88
492 F2 DATA 30,39,00,00,00,02,08,00,00,07
493 4Q DATA 66,00,00,36,08,39,00,06,00,8F
494 70 DATA 20,01,66,8E,22,79,00,00,3A,56
495 13 DATA 3A,29,00,00,00,3C,29,00,00,8A,7C
496 06 DATA 00,54,65,D4,BC,7C,00,87,65,CE
497 72 DATA 8A,7C,00,8C,62,C8,3C,7C,00,96
498 72 DATA 62,C2,60,00,00,00,30,39,00,00
499 4A DATA 30,02,08,00,00,00,07,66,F4,0C,B9
500 11 DATA 00,00,00,00,00,00,2F,86,66,00
501 12 DATA 00,0C,23,FC,00,00,00,01,00,00
502 8P DATA 2F,86,2C,79,00,00,00,3A,46,20,79
503 50 DATA 00,00,3A,56,4E,AE,FF,88,23,FC
504 Y0 DATA 00,00,00,00,00,00,2D,A8,2C,79
505 4C DATA 00,00,3A,3E,20,79,00,00,3A,62
506 77 DATA 43,F9,00,00,32,6E,30,3C,00,20
507 7v DATA 4E,AE,FF,40,0C,B9,00,00,00,00
508 2A DATA 00,00,2F,86,66,00,00,00,0C,23,FC
509 4v DATA 00,00,00,00,00,00,2F,86,0C,B9
510 40 DATA 00,00,00,00,00,00,2F,E2,66,00
511 7C DATA 00,0C,23,FC,00,00,00,01,00,00
512 77 DATA 2F,E2,20,39,00,00,2F,86,22,39
513 77 DATA 00,00,2F,E2,82,80,6C,00,00,0E
514 41 DATA 23,C1,00,00,2F,86,23,C0,00,00
515 4g DATA 2F,E2,20,39,00,00,3A,78,80,B9
516 40 DATA 00,00,2F,E2,65,00,01,04,2C,79
517 8A DATA 00,00,3A,42,24,3C,00,00,03,EE
518 32 DATA 22,3C,00,00,2E,EA,4E,AE,FF,E2
519 47 DATA 23,C0,00,00,3A,86,67,00,00,24
520 17 DATA 23,FC,41,4E,45,20,00,00,00,20
521 4A DATA 22,00,24,3C,00,00,00,00,26,3C
522 24 DATA 00,00,00,04,4E,AE,FF,D0,67,00
523 62 DATA 00,A0,20,39,00,00,2F,E2,90,B9
524 3C DATA 00,00,2F,86,60,7C,00,01,33,00
525 4v DATA 00,00,00,00,22,39,00,00,3A,86
526 47 DATA 24,3C,00,00,00,00,26,3C,00,00
527 2a DATA 00,02,4E,AE,FF,D0,67,00,00,70
528 17 DATA 22,39,00,00,3A,86,24,3C,00,00
529 2p DATA 32,8E,26,3C,00,00,00,20,4E,AE
530 4C DATA FF,D0,67,00,00,00,56,4E,B9,00
531 7v DATA 2E,20,24,39,00,00,2F,86,94,BC
532 42 DATA 00,00,00,01,C4,FC,02,80,D4,B9
533 2P DATA 00,00,3A,4A,26,39,00,00,2E,E2
534 40 DATA 96,B9,00,00,2F,86,6C,BC,00,00
535 77 DATA 00,01,C6,FC,02,80,22,39,00,00
536 47 DATA 3A,86,4E,AE,FF,D0,67,00,00,16
537 4E DATA 4E,B9,00,00,00,2A,F2,22,39,00,00
538 2A DATA 3A,86,4E,AE,FF,DC,60,00,02,F6
539 47 DATA 20,79,00,00,2D,24,2C,79,00,00
540 77 DATA 3A,46,4E,AE,FF,DC,60,2C,79,00,00
541 77 DATA 3A,42,22,39,00,00,3A,86,4E,AE

542 fr DATA FF,DC,60,00,F2,D2,20,79,00,00
543 FJ DATA 2D,B4,2C,79,00,00,3A,46,4E,AE
544 JJ DATA FF,A0,60,00,F2,BE,2C,79,00,00
545 47 DATA 3A,3E,43,F9,00,00,3A,98,20,3C
546 7p DATA 00,00,00,10,4E,AE,FE,AA,43,F9
547 3e DATA 00,00,3A,98,42,80,42,81,34,3C
548 qD DATA 00,1F,36,3C,00,1F,4E,AE,FE,CE
549 nY DATA 4E,B9,00,00,27,A6,60,00,F2,88
550 Pa DATA 20,79,00,00,3B,10,61,00,00,28
551 0E DATA 20,79,00,00,3B,14,61,00,00,1E
552 4r DATA 20,79,00,00,3B,18,61,00,00,14
553 4A DATA 20,79,00,00,3B,1C,61,00,00,0A
554 ab DATA 61,00,12,D8,60,00,F2,58,20,3C
555 k8 DATA 00,00,00,1F,22,10,24,10,48,42
556 xg DATA C2,BC,00,00,FF,FF,04,BC,00,00
557 49 DATA FF,FF,42,83,42,84,2E,30,00,00
558 8R DATA 00,0F,0F,01,67,00,00,10,46,47
559 La DATA CE,7C,00,0F,0F,03,46,47,CE,7C
560 Hn DATA 00,0F,51,CF,FF,EA,2E,3C,00,00
561 4X DATA 00,0F,0F,02,67,00,00,10,46,47
562 2h DATA CE,70,00,0F,0F,04,46,47,CE,7C
563 3s DATA 00,0F,51,CF,FF,EA,4B,43,86,84
564 0k DATA 20,C3,51,C8,FF,A6,4E,75,20,79
565 7u DATA 00,00,3B,10,61,00,00,28,20,79
566 46 DATA 00,00,3B,14,61,00,00,1E,20,79
567 xv DATA 00,00,3B,18,61,00,00,14,20,79
568 4U DATA 00,00,3B,1C,61,00,00,0A,61,00
569 18 DATA 12,44,60,00,F1,C4,22,8D,D3,FC
570 4N DATA 00,00,00,80,20,3C,00,00,00,0F
571 EG DATA 22,10,24,21,22,81,20,C2,51,08
572 Tq DATA FF,F6,4E,75,20,79,00,00,3B,14
573 2p DATA 61,00,00,28,20,79,00,00,3B,14
574 ep DATA 61,00,00,1E,20,79,00,00,3B,18
575 05 DATA 61,00,00,14,20,79,00,00,3B,1C
576 7n DATA 61,00,00,0A,61,00,11,F8,60,00
577 Rd DATA F1,78,43,F9,00,00,00,00,24,48
578 3s DATA 2C,3C,00,00,00,1F,2E,3C,00,00
579 0Y DATA 00,1F,42,83,20,79,00,00,18,00
580 Ae DATA 67,00,00,10,46,47,CE,7C,00,1F
581 12 DATA 0F,C3,46,47,CE,7C,00,1F,51,CF
582 qw DATA FF,E8,22,C3,51,CE,FF,D8,43,F9
583 0e DATA 00,00,00,00,20,4A,2C,3C,00,00
584 6S DATA 00,1F,20,D9,51,CE,FF,FC,4E,75
585 FI DATA 20,79,00,00,3A,62,30,3C,00,1F
586 0X DATA 32,3C,00,00,3A,01,36,01,2C,79
587 0b DATA 00,00,3A,3E,4E,AE,FE,E0,20,79
588 4F DATA 00,00,3A,62,30,3C,00,1E,32,3C
589 ea DATA 00,03,34,01,36,01,4E,AE,FE,E0
590 fm DATA 2C,79,00,00,3A,46,23,FC,00,00
591 T2 DATA 31,5A,00,00,2D,A8,41,F9,00,00
592 cb DATA 2D,96,4E,AE,FF,34,23,C0,00,00
593 1L DATA 3A,56,20,40,24,3C,00,32,00,00
594 KM DATA 3A,6A,23,88,00,32,00,00,3A,6E
595 dE DATA 22,68,00,32,30,3C,00,01,2C,79
596 Vx DATA 00,00,3A,3E,4E,AE,FE,AA,22,79
597 81 DATA 00,00,3A,6A,30,3C,00,01,4E,AE
598 ep DATA FE,A4,22,79,00,00,3A,6A,30,3C
599 4B DATA 00,00,32,3C,00,00,3A,3C,00,DF
600 na DATA 36,3C,00,00,4E,AE,FE,CE,22,79
601 MQ DATA 00,00,3A,6A,30,3C,00,00,32,3C
602 4v DATA 00,55,34,3C,00,39,36,3C,00,61
603 1v DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
604 7Y DATA 30,3C,00,59,32,3C,00,55,34,3C
605 f9 DATA 00,85,36,3C,00,61,4E,AE,FE,CE
606 01 DATA 22,79,00,00,3A,6A,30,3C,00,A9
607 qa DATA 32,3C,00,55,34,3C,00,DF,36,3C
608 H1 DATA 00,61,4E,AE,FE,CE,22,79,00,00
609 CK DATA 3A,6A,30,3C,00,00,32,3C,00,61
610 0t DATA 34,3C,00,0F,36,3C,00,68,4E,AE
611 dm DATA FE,CE,22,79,00,00,3A,6A,30,3C
612 7x DATA 00,00,32,3C,00,69,34,3C,00,79
613 4J DATA 36,3C,00,75,4E,AE,FE,CE,22,79
614 4p DATA 00,00,3A,6A,30,3C,00,9D,32,3C
615 4E DATA 00,69,34,3C,00,DF,36,3C,00,75
616 48 DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
617 7I DATA 30,3C,00,00,32,3C,00,75,34,3C
618 76 DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
619 7e DATA 4B,F9,00,00,3C,4F,3A,3C,00,13
620 7t DATA 3C,3C,00,23,3E,3C,00,4F,4E,B9
621 F1 DATA 00,00,04,18,4E,AE,FE,CE,22,79
622 4N DATA 3A,3C,00,04,3C,3C,00,5F,3E,3C
623 2T DATA 00,5D,4E,B9,00,00,04,18,4E,B9
624 6B DATA 00,00,3C,62,3A,3C,00,06,3C,FC
625 wt DATA 00,3F,3E,3C,00,71,4E,B9,00,00

626 K3 DATA 04,18,4B,F9,00,00,3B,9F,3A,3C
627 pa DATA 00,06,3C,3C,00,58,3E,3C,00,91
628 78 DATA 4E,B9,00,00,04,18,30,3C,00,02
629 N1 DATA 3B,3C,00,04,3A,3C,00,87,3C,3C
630 RU DATA 00,8C,3E,3C,00,96,4E,B9,00,00
631 ok DATA 04,88,30,39,00,00,32,1E,08,00
632 vq DATA 00,07,66,00,00,36,08,39,00,06
633 3M DATA 00,BF,EO,01,66,EB,22,79,00,00
634 AQ DATA 3A,56,3A,29,00,0E,3C,29,00,0C
635 Ey DATA BA,7C,00,54,65,D4,BC,7C,00,87
636 9M DATA 65,CE,BA,7C,00,8C,62,C8,BC,7C
637 rF DATA 00,96,62,C2,60,00,00,0E,30,39
638 xw DATA 00,00,32,1E,08,00,00,07,66,P4
639 nh DATA 2C,79,00,00,3A,46,20,79,00,00
640 DY DATA 3A,56,4E,AE,FF,B8,23,FC,00,00
641 w1 DATA 00,00,00,00,2D,AB,2C,79,00,00
642 Yv DATA 3A,3E,20,79,00,00,3A,62,43,F9
643 4U DATA 00,00,32,6E,30,3C,00,20,4E,AE
644 k2 DATA FF,40,4A,B9,00,00,31,A2,62,00
645 LM DATA 00,0C,23,FC,02,80,00,01,00,00
646 ry DATA 31,A2,4A,B9,00,00,31,FE,62,00
647 NO DATA 00,0C,23,FC,00,00,00,01,00,00
648 B1 DATA 31,FE,4A,B9,00,00,32,5A,63,00
649 JO DATA 01,00,20,39,00,00,31,A2,D0,B9
650 x0 DATA 00,00,32,5A,90,BC,00,00,00,01
651 Jh DATA B0,B9,00,00,3A,78,62,00,00,EE
652 3E DATA 22,39,00,00,31,FE,D2,B9,00,00
653 SQ DATA 32,5A,92,BC,00,00,00,01,B2,B9
654 hr DATA 00,00,3A,78,62,00,00,D2,B2,80
655 o8 DATA 67,00,00,C2,D0,B9,00,00,31,FE
656 aR DATA 62,00,00,4C,20,39,00,00,32,5A
657 de DATA 00,FC,00,40,90,BC,00,00,00,01
658 pb DATA 22,39,00,00,31,A2,92,BC,00,00
659 OF DATA 00,01,C2,FC,02,80,00,D2,B9,00,00
660 JQ DATA 3A,4A,20,41,22,39,00,00,31,FE
661 zV DATA 92,BC,00,00,00,01,C2,FC,02,80
662 40 DATA D2,B9,00,00,3A,4A,22,39,00,00
663 ev DATA 51,C8,FF,FC,60,00,00,6E,20,39
664 FK DATA 00,00,31,A2,80,B9,00,00,31,FE
665 PF DATA 62,AB,20,39,00,00,32,5A,0C,FC
666 95 DATA 00,AD,90,BC,00,00,00,01,22,39
667 CR DATA 00,00,31,A2,D2,B9,00,00,32,5A
668 6a DATA 92,BC,00,00,00,01,C2,FC,02,80
669 ya DATA D2,B9,00,00,3A,4A,22,39,00,00
670 LZ DATA 00,00,31,FE,D2,B9,00,00,32,5A
671 9f DATA 92,BC,00,00,00,01,C2,FC,02,80
672 xJ DATA D2,B9,00,00,3A,4A,22,41,D1,FC
673 Tt DATA 00,00,00,04,D3,FC,00,00,00,04
674 HA DATA 23,20,51,C8,FF,FC,4E,B9,00,00
675 yM DATA 27,AE,60,00,ED,AD,2C,79,00,00
676 00 DATA 3A,46,20,79,00,00,2D,84,4E,AE
677 ZV DATA FF,AD,60,00,ED,8C,41,F9,00,00
678 t6 DATA 2D,96,2C,79,00,00,3A,46,4E,AE
679 FW DATA FF,34,4A,80,67,00,ED,76,23,C0
680 UK DATA 00,00,3A,56,20,40,23,EB,00,32
681 Ww DATA 00,00,3A,6A,22,68,00,32,30,3C
682 dt DATA 00,01,2C,79,00,00,3A,3E,4E,AE
683 Fe DATA FE,AA,22,79,00,00,3A,6A,30,3C
684 6A DATA 00,01,4E,AE,FE,AA,22,79,00,00
685 Tt DATA 3A,6A,42,40,42,41,3A,3C,00,DF
686 Da DATA 36,3C,00,DF,4E,AE,FE,CE,23,F9
687 6U DATA 00,00,3A,6A,60,3C,00,3E,4E,AE
688 ER DATA 00,00,3B,BD,3A,3C,00,0C,3C,3C
689 of DATA 00,40,3E,3C,00,50,4E,B9,00,00
690 Og DATA 04,18,4B,F9,00,00,3B,C9,3A,3C
691 pv DATA 00,0B,3C,3C,00,44,3E,3C,00,5A
692 54 DATA 4E,B9,00,00,04,18,4B,F9,00,00
693 K4 DATA 3B,D4,3A,3C,00,0F,3C,3C,00,34
694 4L DATA 3E,3C,00,69,4E,B9,00,00,04,18
695 N8 DATA 4B,F9,00,00,3B,E3,3A,3C,00,0F
696 8r DATA 3C,3C,00,34,3E,3C,00,73,4E,B9
697 At DATA 00,00,04,18,4B,F9,00,00,3B,FE
698 93 DATA 3A,3C,00,11,3C,3C,00,2C,3E,3C
699 3p DATA 00,82,4E,B9,00,00,04,18,4B,F9
700 m0 DATA 00,00,3B,9F,3A,3C,00,06,3C,3C
701 Hz DATA 00,58,3E,3C,00,96,4E,B9,00,00
702 Yb DATA 04,18,30,3C,00,02,3B,3C,00,54
703 ms DATA 3A,3C,00,8C,3C,3C,00,8C,3E,3C
704 sp DATA 00,9B,4E,B9,00,00,04,88,08,39
705 IV DATA 00,06,00,BF,EO,01,66,FE,22,79
706 Uy DATA 00,00,3A,56,3A,29,00,0E,3C,29
707 Jo DATA 00,0C,BA,7C,00,54,65,E2,BC,7C
708 mo DATA 00,8C,65,DC,BA,7C,00,8C,62,D6
709 us DATA BC,7C,00,9B,62,D0,2C,79,00,00

710 W DATA 3A,46,20,79,00,00,3A,56,4E,AE
711 JM DATA FF,8a,60,00,EC,3E,41,F9,00,00
712 3E DATA 2D,96,23,FC,00,00,30,42,00,00
713 V DATA 2D,AB,2C,79,00,00,3A,46,4E,AE
714 1a DATA FF,34,23,C0,00,00,3A,56,20,40
715 4c DATA 23,EB,00,32,00,00,3A,62,23,EB
716 Hq DATA 00,32,00,00,3A,6E,22,68,00,32
717 32 DATA 30,3C,00,01,2C,79,00,00,3A,3E
718 1f DATA 4E,AE,FE,AA,22,79,00,00,3A,6A
719 a DATA 30,3C,00,01,4E,AE,FE,AA,20,79
720 GS DATA 00,00,3A,62,30,3C,00,1F,32,3C
721 ey DATA 00,0D,3A,01,36,01,2C,79,00,00
722 ap DATA 3A,3E,4E,AE,FE,EO,20,79,00,00
723 m DATA 3A,62,30,3C,00,1E,32,3C,00,03
724 40 DATA 34,01,36,01,4E,AE,FE,EO,22,79
725 yQ DATA 00,00,3A,6A,30,3C,00,00,32,3C
726 zZ DATA 00,00,3A,3C,00,00,3B,39,3A,3C
727 4y DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
728 EK DATA 30,3C,00,00,32,3C,00,5F,34,3C
729 1a DATA 00,35,36,3C,00,06,4E,AE,FE,CE
730 Jh DATA 22,79,00,00,3A,6A,30,3C,00,59
731 1 DATA 32,3C,00,5F,34,3C,00,85,36,3C
732 Wg DATA 00,6B,4E,AE,FE,CE,22,79,00,00
733 og DATA 3A,6A,30,3C,00,00,3B,39,3A,3C
734 2L DATA 34,3C,00,DF,36,3C,00,6B,4E,AE
735 1a DATA FE,CE,22,79,00,00,3A,6A,30,3C
736 4c DATA 00,00,32,3C,00,6B,34,3C,00,DF
737 K DATA 36,3C,00,DF,4E,AE,FE,CE,4B,F9
738 9 DATA 00,00,3C,1A,3A,3C,00,24,3C,3C
739 4E DATA 00,1F,3E,3C,00,59,4E,B9,00,00
740 2w DATA 04,18,4B,F9,00,00,3A,62,43,F9
741 4y DATA 00,04,3C,3C,00,5F,3E,3C,00,66
742 1s DATA 4E,B9,00,00,04,18,4B,F9,00,00
743 2L DATA 3B,9F,3A,3C,00,06,3C,00,58
744 1a DATA 3E,3C,00,7D,4E,B9,00,00,04,18
745 DS DATA 30,3C,00,02,38,3C,00,54,3A,3C
746 5X DATA 00,73,3C,3C,00,8C,3E,3C,00,82
747 2v DATA 4E,B9,00,00,04,88,08,39,00,06
748 38 DATA 00,BF,EO,01,66,FE,22,79,00,00
749 1H DATA 3A,56,3A,29,00,0E,3C,29,00,0C
750 3H DATA BA,7C,00,54,65,E2,BC,7C,00,73
751 9 DATA 65,DC,BA,7C,00,8C,62,D6,BC,7C
752 2A DATA 00,82,62,D0,2C,79,00,00,3A,46
753 3q DATA 20,79,00,00,3A,56,4E,AE,FF,38
754 47 DATA 0C,B9,00,00,00,00,00,00,30,EA
755 4s DATA 66,00,00,0C,23,FC,00,00,00,01
756 1 DATA 00,00,30,EA,20,39,00,00,3A,78
757 1D DATA B0,B9,00,00,30,EA,62,00,00,00
758 4N DATA 23,F9,00,00,3A,78,00,00,30,EA
759 0 DATA 0C,B9,00,00,00,00,00,00,31,46
760 3x DATA 66,00,00,0C,23,FC,00,00,00,00
761 FN DATA 00,00,31,46,20,39,00,00,3A,78
762 1t DATA B0,B9,00,00,31,46,62,00,00,0C
763 4w DATA 23,F9,00,00,3A,78,00,00,31,46
764 2s DATA 20,39,00,00,30,EA,22,39,00,00
765 2A DATA 31,46,B2,80,64,00,00,0E,23,C1
766 SJ DATA 00,00,30,EA,23,C0,00,00,31,46
767 7 DATA 23,F9,00,00,30,EA,00,00,3A,6E
768 1t DATA 41,F9,00,00,2D,96,23,FC,00,00
769 1a DATA 00,00,00,00,2D,AB,2C,79,00,00
770 1a DATA 3A,46,4E,AE,FF,34,23,C0,00,00
771 XL DATA 3A,56,20,40,23,EB,00,32,00,00
772 DATA 3A,6A,23,EB,00,32,00,00,3A,6E
773 1a DATA 22,68,00,32,22,79,00,00,3A,6A
774 2x DATA 30,3C,00,01,2C,79,00,00,3A,3E
775 3E DATA 4E,AE,FE,AA,20,79,00,00,3A,62
776 4c DATA 43,F9,00,00,32,6E,30,3C,00,00
777 1a DATA 4E,AE,FF,40,22,79,00,00,3A,6A
778 4c DATA 30,3C,00,01,4E,AE,FE,AA,22,79
779 2L DATA 00,00,3A,6A,30,3C,00,00,32,3C
780 4c DATA 00,00,3A,3C,00,0F,36,3C,00,3F
781 2L DATA 4E,AE,FE,CE,22,79,00,00,3A,46
782 2q DATA 20,79,00,00,3A,6A,43,F9,00,00
783 12 DATA 35,9E,30,3C,00,58,32,3C,00,57
784 4H DATA 4E,AE,FF,EE,2C,79,00,00,3A,1E
785 3c DATA 4B,F9,00,00,3C,2E,3A,3C,00,0E
786 1f DATA 3C,3C,00,34,3E,3C,00,4F,4E,B9
787 1a DATA 00,00,04,18,4B,F9,00,00,3B,9F
788 61 DATA 3A,3C,00,06,3C,3C,00,58,3E,3C
789 5 DATA 00,7D,4E,B9,00,00,04,18,4B,F9
790 12 DATA 00,00,3C,3D,3A,3C,00,03,3C,3C
791 55 DATA 00,63,3E,3C,00,5D,4E,B9,00,00
792 3 DATA 04,18,30,3C,00,02,3B,3C,00,54
793 2G DATA 3A,3C,00,73,3C,3C,00,8C,3E,3C

794 XL DATA 00,82,4E,B9,00,00,04,88,30,3C
795 EE DATA 00,10,22,79,00,00,3A,6A,4E,AE
796 4R DATA FE,AA,22,79,00,00,3A,6A,30,3C
797 4c DATA 00,5F,32,3C,00,14,34,3C,00,7E
798 4c DATA 36,3C,00,33,4E,AE,FE,CE,30,3C
799 3c DATA 00,02,3B,3C,00,5E,3A,3C,00,13
800 eM DATA 3C,3C,00,7F,3E,3C,00,34,4E,B9
801 12 DATA 00,00,04,88,30,3C,00,03,38,3C
802 WM DATA 00,5D,3A,3C,00,12,3C,3C,00,80
803 4c DATA 3E,3C,00,35,4E,B9,00,00,04,88
804 1a DATA 30,3C,00,04,38,3C,00,5C,3A,3C
805 45 DATA 00,11,3C,3C,00,81,3E,3C,00,36
806 KZ DATA 4E,B9,00,00,04,88,23,F9,00,00
807 47 DATA 3B,10,00,00,00,00,23,F9,00,00
808 1W DATA 3B,14,00,00,00,04,23,F9,00,00
809 4w DATA 3B,18,00,00,00,08,23,F9,00,00
810 4c DATA 3B,1C,00,00,00,0C,23,F9,00,00
811 K DATA 3B,20,00,00,00,10,23,F9,00,00
812 1N DATA 3A,92,00,00,3A,8A,2A,7C,00,DF
813 4y DATA F0,00,23,F8,00,6C,00,00,27,6F
814 FM DATA 21,FC,00,00,26,AC,00,6C,00,79
815 61 DATA 00,01,00,00,3A,96,66,00,00,54
816 1 DATA 20,39,00,DF,F0,04,0C,BC,00,01
817 1a DATA FF,00,B0,BC,00,01,00,00,66,EC
818 4c DATA 2C,79,00,00,3A,3E,41,F9,00,00
819 4c DATA 3A,98,22,79,00,00,3A,6A,42,80
820 1a DATA 42,81,24,3C,00,00,00,5F,26,3C
821 0 DATA 00,00,00,14,28,3C,00,00,00,20
822 FF DATA 2A,3C,00,00,00,20,20,3C,00,00
823 31 DATA 00,00,4E,AE,FD,D8,42,79,00,00
824 48 DATA 3A,96,08,39,00,00,0C,BF,EO,01
825 4c DATA 66,98,22,79,00,00,3A,56,3A,29
826 1a DATA 00,0E,3C,29,00,00,BA,7C,00,58
827 4R DATA 65,00,00,1E,BC,7C,00,00,57,65,00
828 sv DATA 00,16,BA,7C,00,5F,62,00,00,0E
829 xy DATA BC,7C,00,5E,62,00,00,06,60,00
830 76 DATA 0A,06,BA,7C,00,80,65,00,00,22
831 2L DATA 7C,7C,00,57,65,00,0C,1A,BA,7C
832 4B DATA 00,87,62,00,00,12,BC,7C,00,5E
833 7S DATA 62,00,00,0A,60,00,0A,22,60,00
834 01 DATA FF,40,BA,7C,00,5A,65,00,FF,38
835 11 DATA BC,7C,00,73,65,00,FF,30,BA,7C
836 2N DATA 00,8C,62,00,FF,28,BC,7C,00,82
837 3a DATA 62,00,FF,20,2A,7C,00,DF,F0,00
838 4c DATA 21,F9,00,00,27,68,00,6C,23,F9
839 3a DATA 00,00,00,00,00,00,3B,10,23,F9
840 17 DATA 00,00,00,04,00,00,3B,14,23,F9
841 4e DATA 00,00,00,08,00,00,3B,18,23,F9
842 15 DATA 00,00,00,0C,00,00,3B,1C,23,F9
843 1 DATA 00,00,00,10,00,00,3B,20,2C,79
844 4c DATA 00,00,3A,46,20,79,00,00,3A,56
845 17 DATA 4E,AE,FF,B8,23,F9,00,00,3A,66
846 3Q DATA 00,00,3A,6E,23,FC,00,00,00,00
847 3H DATA 00,00,2D,AB,60,00,0E,66,30,39
848 4c DATA 00,00,3A,76,BC,7C,00,01,66,00
849 63 DATA 00,06,60,00,0E,D4,04,B9,00,00
850 4c DATA 02,80,00,00,3B,10,04,B9,00,00
851 BC DATA 02,80,00,00,3B,14,04,B9,00,00
852 1t DATA 02,80,00,00,3B,18,04,B9,00,00
853 4c DATA 02,80,00,00,3B,1C,04,B9,00,00
854 4c DATA 02,80,00,00,3B,20,04,79,00,01
855 21 DATA 00,00,3A,76,4E,B9,00,00,2A,42
856 PX DATA 4E,B9,00,00,27,AE,60,00,0E,6A
857 12 DATA 30,39,00,00,3A,76,BC,7C,00,00
858 1F DATA 3A,78,66,00,00,06,60,00,0E,76
859 2A DATA 06,B9,00,00,02,80,00,00,3B,10
860 4Z DATA 06,B9,00,00,02,80,00,00,3B,14
861 4y DATA 06,B9,00,00,02,80,00,00,3B,18
862 23 DATA 06,B9,00,00,02,80,00,00,3B,1C
863 4c DATA 06,B9,00,00,02,80,00,00,3B,20
864 Gq DATA 06,79,00,01,00,00,3A,76,4E,B9
865 46 DATA 00,00,2A,42,4E,B9,00,00,27,AE
866 1C DATA 60,00,0E,2C,42,80,30,39,00,00
867 4j DATA 3A,72,C0,FC,00,02,20,40,D1,FC
868 4c DATA 00,00,32,8E,30,10,00,7C,0F,00
869 3c DATA EO,58,DD,7C,00,01,00,7C,00,0F
870 4c DATA E1,58,02,50,00,FF,81,50,4E,B9
871 1t DATA 00,00,27,6C,4E,B9,00,00,05,02
872 4c DATA 60,00,B5,70,42,80,30,39,00,00
873 04 DATA 3A,72,00,FC,00,02,20,40,D1,FC
874 4A DATA 00,00,32,8E,30,10,00,7C,0F,00

Listing 1. (Fortsetzung)

875 es DATA E0,5B,90,7C,00,01,00,7C,00,0F
876 Me DATA E1,58,02,50,00,FF,81,50,4E,B9
877 Pa DATA 00,00,27,6C,4E,B9,00,00,05,C2
878 DM DATA 60,00,E5,B4,42,80,30,39,00,00
879 tA DATA 3A,72,0C,FC,00,02,20,40,D1,FC
880 fy DATA 00,00,32,8E,30,10,00,7C,00,0F
881 dC DATA E8,58,02,50,00,0F,81,50,4E,B9
882 uI DATA E9,58,02,50,0F,0F,81,50,4E,B9
883 vS DATA 00,00,27,6C,4E,B9,00,00,05,C2
884 nJ DATA 60,00,E5,78,42,80,30,39,00,00
885 eG DATA 3A,72,0C,FC,00,02,20,40,D1,FC
886 lA DATA 00,00,32,8E,30,10,00,7C,00,0F
887 Ca DATA E8,58,90,7C,00,01,00,7C,00,0F
888 OJ DATA E9,58,02,50,0F,0F,81,50,4E,B9
889 bB DATA 00,00,27,6C,4E,B9,00,00,05,C2
890 Rm DATA 60,00,E5,3C,42,80,30,39,00,00
891 5M DATA 3A,72,0C,FC,00,02,20,40,D1,FC
892 DW DATA 00,00,32,8E,30,10,00,7C,00,0F
893 fL DATA D0,70,00,01,00,70,00,0F,02,50
894 VW DATA 0F,0F,81,50,4E,B9,00,00,27,6C
895 lH DATA 4E,B9,00,00,05,02,60,79,00,00
896 QD DATA 42,80,30,39,00,00,3A,72,0C,FC
897 iV DATA 00,02,20,40,D1,FC,00,00,32,8E
898 Jj DATA 30,10,00,7C,00,00,0F,90,70,00,00
899 sJ DATA 00,7C,00,0F,02,50,0F,0F,81,50
900 bF DATA 4E,B9,00,00,27,6C,4E,B9,00,00
901 vO DATA 05,02,60,00,E4,00,20,79,00,00
902 JD DATA 3A,46,41,89,00,00,20,C6,4E,AE
903 4k DATA FF,34,23,00,00,00,3A,56,20,40
904 Bt DATA 23,E8,00,32,00,00,3A,6A,23,E8
905 KI DATA 00,32,00,00,3A,6E,22,80,00,32
906 sS DATA 30,3C,00,01,20,79,00,00,3A,3E
907 jV DATA 4E,AE,FE,AA,20,79,00,00,3A,62
908 nr DATA 30,3C,00,1F,32,30,00,00,3A,01
909 tC DATA 36,01,4E,AE,FE,00,20,79,00,00
910 Qo DATA 3A,62,30,30,00,1E,32,30,00,03
911 QP DATA 3A,01,36,01,4E,AE,FE,00,22,79
912 Dn DATA 00,00,3A,6A,30,3C,00,01,4E,AE
913 OH DATA FE,AA,22,79,00,00,3A,6A,42,40
914 kS DATA 42,41,34,3C,00,0F,36,3C,00,00
915 DM DATA 4E,AE,FE,CE,42,40,32,30,00,00
916 KE DATA 3A,3C,00,36,36,30,00,18,22,79
917 m4 DATA 00,00,3A,6A,4E,AE,FE,CE,30,3C
918 IG DATA 00,3A,32,3C,00,00,3A,3C,00,8F
919 CG DATA 36,3C,00,18,22,79,00,00,3A,6A
920 TS DATA 4E,AE,FE,CE,42,40,32,30,00,19
921 A0 DATA 3A,3C,00,0F,36,3C,00,20,22,79
922 r9 DATA 00,00,3A,6A,4E,AE,FE,CE,30,3C
923 YQ DATA 00,03,42,44,42,43,3C,00,8F
924 Ay DATA 3E,3C,00,20,4E,B9,00,00,04,88
925 lS DATA 4B,F9,00,00,3C,03,3A,3C,00,10
926 fo DATA 3C,3C,00,08,3E,30,00,08,4E,B9
927 eU DATA 00,00,04,1B,4B,F9,00,00,3B,9F
928 kV DATA 3A,3C,00,06,30,3C,00,30,3E,30
929 Ba DATA 00,25,4E,B9,00,00,04,18,30,30
930 Wt DATA 00,02,38,3C,00,20,3A,3C,00,1B
931 sY DATA 3C,3C,00,64,3E,3C,00,2A,4E,B9
932 pv DATA 00,00,04,88,30,39,00,00,30,52
933 FF DATA 08,00,00,07,66,00,00,36,08,39
934 XI DATA 00,06,00,BF,E0,01,66,E8,22,79
935 Bf DATA 00,00,3A,56,3A,29,00,00,30,29
936 lE DATA 00,00,BA,7C,00,20,65,04,BC,7C
937 oR DATA 00,1B,65,CE,BA,7C,00,64,62,C8
938 DO DATA BC,7C,00,2A,62,C2,60,00,00,0E
939 jG DATA 30,39,00,00,30,52,00,00,00,07
940 vL DATA 66,F4,2C,79,00,00,3A,46,20,79
941 Pu DATA 00,00,3A,56,4E,AE,FF,B8,0C,B9
942 xH DATA 00,00,00,00,00,00,30,8E,66,00
943 9A DATA 00,00,23,FC,00,00,00,01,00,00
944 3a DATA 30,8E,20,39,00,00,3A,78,B0,B9
945 DE DATA 00,00,3D,8E,64,00,00,00,23,F9
946 XN DATA 00,00,3A,78,00,00,30,8E,20,39
947 nW DATA 00,00,30,8E,33,00,00,00,3A,76
948 Kw DATA 90,BC,00,00,00,01,00,FC,02,80
949 AG DATA D0,B9,00,00,3A,4A,23,C0,00,00
950 b2 DATA 3B,10,00,BC,00,00,00,80,23,00
951 Bp DATA 00,00,3B,14,00,BC,00,00,00,80
952 eV DATA 23,C0,00,00,3B,1B,00,BC,00,00
953 2n DATA 00,80,23,C0,00,00,3B,1C,00,BC
954 9P DATA 00,00,00,80,23,C0,00,00,3B,20
955 aI DATA 20,79,00,00,3A,3E,20,79,00,00
956 vS DATA 3A,62,43,F9,00,00,32,6E,30,3C
957 bD DATA 00,20,4E,AE,FF,04,4E,B9,00,00
958 4W DATA 2A,42,4E,B9,00,00,27,4E,60,00

959 Jy DATA E2,8C,23,FC,00,00,00,64,00,00
960 lK DATA 30,8E,20,3C,00,31,30,30,01,E8
961 OT DATA 23,00,00,00,30,96,23,00,00,C2
962 V8 DATA 30,9C,2C,79,00,00,3A,46,41,F9
963 lA DATA 00,00,2D,C6,4E,AE,FF,3A,23,C0
964 4u DATA 00,00,3A,56,20,40,23,E8,00,32
965 ek DATA 00,00,3A,6A,23,E8,00,32,00,00
966 yb DATA 3A,6E,22,68,00,32,30,3C,00,01
967 V8 DATA 2C,79,00,00,3A,3E,4E,AE,FE,AA
968 QT DATA 20,79,00,00,3A,62,30,3C,00,1F
969 rQ DATA 32,3C,00,0D,34,01,36,01,4E,AE
970 Sl DATA FE,80,20,79,00,00,3A,62,30,3C
971 Mv DATA 00,1E,32,30,00,03,34,01,36,01
972 EY DATA 4E,AE,FE,80,22,79,00,00,3A,6A
973 kP DATA 30,3C,00,01,4E,AE,FE,AA,22,79
974 YK DATA 00,00,3A,6A,42,40,42,41,3A,3C
975 7K DATA 00,8F,36,3C,00,00,0C,4E,AE,FE,CE
976 7L DATA 42,40,32,3C,00,0D,34,3C,00,36
977 8C DATA 36,3C,00,1B,22,79,00,00,3A,6A
978 tE DATA 4E,AE,FE,CE,30,3C,00,5A,32,3C
979 oE DATA 00,0D,34,3C,00,0F,0C,30,00,18
980 Fg DATA 22,79,00,00,3A,6A,4E,AE,FE,CE
981 lI DATA 42,40,32,3C,00,19,3A,3C,00,8F
982 Pq DATA 36,3C,00,2C,22,79,00,00,3A,01
983 mW DATA 4E,AE,FE,CE,30,3C,00,03,42,44
984 oP DATA 42,45,3C,3C,00,8F,3E,3C,00,2C
985 28 DATA 4E,B9,00,00,04,88,4B,F9,00,00
986 RO DATA 30,40,3A,3C,00,0F,0C,30,00,0C
987 Bp DATA 3E,3C,00,08,4E,B9,00,00,04,18
988 Kx DATA 4B,F9,00,00,3B,9F,3A,3C,00,06
989 oE DATA 3C,3C,00,30,3E,60,00,00,25,4E,B9
990 CJ DATA 00,00,04,18,30,3C,00,02,38,30
991 CL DATA 00,2C,3A,3C,00,1B,3C,3C,00,64
992 Cy DATA 3E,3C,00,2A,4E,B9,00,00,04,88
993 lY DATA 30,39,00,00,30,52,08,00,00,07
994 Vv DATA 66,00,00,36,08,39,00,00,06,0F
995 OT DATA E0,01,66,E8,22,79,00,00,3A,56
996 QA DATA 3A,29,00,30,8E,60,00,00,0C,BA,7C
997 T9 DATA 00,2C,65,D4,BC,7C,00,1B,65,CE
998 W0 DATA BA,7C,00,64,62,C8,BC,7C,00,2A
999 Pe DATA 62,C2,60,00,00,0E,30,39,00,00
1000 37 DATA 30,52,08,00,00,07,66,F4,20,79
1001 PN DATA 00,00,3A,46,20,79,00,00,3A,56
1002 Kk DATA 4E,AE,FF,88,0C,B9,00,00,00,00
1003 7K DATA 00,00,30,8E,66,00,00,00,23,FC
1004 er DATA 00,00,00,01,00,00,30,8E,23,F9
1005 XI DATA 00,00,30,8E,00,00,3A,78,2C,79
1006 oq DATA 00,00,3A,3E,20,79,00,00,3A,62
1007 TZ DATA 43,F9,00,00,32,6E,30,3C,00,20
1008 lY DATA 4E,AE,FF,40,23,FC,00,00,00,01
1009 JA DATA 00,00,30,8E,20,3C,00,30,30,31
1010 9k DATA E1,88,23,C0,00,00,30,96,23,FC
1011 0e DATA 00,00,30,9C,4B,75,08,39,00,05
1012 mU DATA 00,DF,0F,1F,66,00,00,06,60,00
1013 jB DATA 00,AC,4B,E7,FF,FE,0C,B9,00,00
1014 Es DATA 00,00,3A,3E,42,80,3A,8A,67,00,10
1015 fC DATA 04,89,00,00,00,01,00,00,3A,8E
1016 K4 DATA 60,00,00,88,20,39,00,00,3A,8A
1017 RJ DATA 90,BC,00,00,00,01,00,FC,02,80
1018 HN DATA D0,B9,00,00,3A,4A,23,C0,00,00
1019 19 DATA 3B,10,00,BC,00,00,00,80,23,00
1020 lW DATA 00,00,3B,14,00,BC,00,00,00,80
1021 z2 DATA 23,C0,00,00,3B,18,00,BC,00,00
1022 9u DATA 00,80,23,C0,00,00,3B,1C,00,BC
1023 0W DATA 00,00,00,80,23,C0,00,00,3B,20
1024 Yf DATA 23,FC,00,01,00,00,3A,96,23,F9
1025 0e DATA 00,00,3A,92,00,00,3A,8A,20,39
1026 Uv DATA 00,00,3A,8E,80,B9,00,00,31,4E
1027 g4 DATA 66,00,00,10,23,F9,00,00,30,EA
1028 yJ DATA 00,00,3A,8E,60,00,00,0C,06,B9
1029 Kk DATA 00,00,00,01,00,00,3A,8E,40,DF
1030 l2 DATA 7F,FF,4E,F9,00,00,00,00,2C,79
1031 oa DATA 00,00,3A,3E,42,80,42,81,42,82
1032 oK DATA 42,83,30,39,00,00,3A,72,D0,7C
1033 Qk DATA 00,10,32,10,34,10,36,10,02,7C
1034 HK DATA 0F,00,00,59,C4,70,00,00,FD,E8,5A
1035 tI DATA C6,7C,00,0F,20,79,00,00,3A,62
1036 3d DATA 4E,AE,FE,E0,4E,75,2C,79,00,00
1037 eL DATA 3A,3E,41,F9,00,00,3A,98,22,79
1038 YN DATA 00,00,3A,66,42,80,42,81,34,3C
1039 9j DATA 01,00,36,3C,00,08,3B,3C,00,20
1040 UT DATA 3A,3C,00,20,3C,3C,00,00,4E,AE
1041 jQ DATA FD,D8,4E,B9,00,00,28,C0,22,79
1042 lR DATA 00,00,3A,66,30,3C,00,02,4E,AE

1043 Wv DATA FE,AA,42,86,22,79,00,00,3A,66
1044 V7 DATA 33,46,00,24,33,7C,00,00,00,26
1045 8T DATA 30,06,32,30,00,00,00,4E,AE,FF,0A
1046 eQ DATA 22,79,00,00,3A,66,33,7C,00,00
1047 gW DATA 00,24,33,46,00,26,30,3C,00,00
1048 9M DATA 32,06,4E,AE,FF,0A,DC,7C,00,07
1049 w8 DATA BC,7C,00,E7,66,C2,22,79,00,00
1050 yH DATA 3A,66,30,3C,00,04,4E,AE,FE,AA
1051 9u DATA 22,79,00,00,3A,66,33,7C,00,7D
1052 hz DATA 00,24,33,7C,00,01,00,26,30,30
1053 Bx DATA 00,70,32,3C,00,DF,4E,AE,FF,0A
1054 pJ DATA 22,79,00,00,3A,66,33,7C,00,01
1055 vU DATA 00,24,33,7C,00,70,00,26,30,3C
1056 vv DATA 00,DF,32,3C,00,70,4E,AE,FF,0A
1057 lB DATA 2C,78,00,04,4E,AE,FF,7C,20,3C
1058 YC DATA 00,00,00,1F,2E,3C,00,00,00,1F
1059 r3 DATA 20,07,22,06,4E,B9,00,00,2A,94
1060 Qy DATA 20,07,22,06,00,FC,00,07,02,FC
1061 sV DATA 00,07,D0,BC,00,00,00,01,D2,BC
1062 oE DATA 00,00,00,01,4E,B9,00,00,29,30
1063 e5 DATA 51,0F,FF,D6,51,CE,FF,C0,2C,78
1064 OA DATA 00,04,4E,AE,FF,76,4E,75,4E,AE
1065 G4 DATA FE,38,2A,7C,00,DF,FD,00,20,79
1066 c5 DATA 00,00,3A,5A,22,28,00,08,4E,B9
1067 Uq DATA 00,00,28,FC,22,28,00,0C,4E,B9
1068 KO DATA 00,00,28,FC,22,28,00,10,4E,B9
1069 bJ DATA 00,00,28,FC,22,28,00,14,4E,B9
1070 D6 DATA 00,00,28,FC,4E,AE,FE,32,08,2D
1071 9r DATA 00,0E,00,02,66,FB,2B,41,00,54
1072 lH DATA 3B,7C,00,00,00,66,3B,7C,FF,FF
1073 ae DATA 00,44,3B,7C,FF,FF,00,46,42,6D
1074 ot DATA 00,42,42,6D,00,74,3B,7C,01,0F
1075 nJ DATA 00,40,3B,7C,38,0E,00,58,4E,75
1076 eT DATA 36,00,06,BC,00,00,00,0F,24,30
1077 Ou DATA FC,00,00,00,E6,AA,C2,FC,00,28
1078 az DATA E6,48,08,80,00,00,D2,80,76,28
1079 Ye DATA 20,79,00,00,3A,5A,08,05,00,00
1080 Le DATA 67,08,22,68,00,00,08,D3,01,61,C2
1081 NE DATA 08,05,00,01,67,08,22,68,00,00
1082 TP DATA D3,C1,61,1E,08,05,00,02,67,08
1083 TG DATA 22,68,00,10,D3,C1,61,10,08,05
1084 l9 DATA 00,03,67,08,22,68,00,14,D3,C1
1085 9h DATA 61,02,4E,75,70,05,85,91,D3,C3
1086 uQ DATA 51,C8,FF,FA,4E,75,0C,B9,00,00
1087 8d DATA 00,01,00,3A,92,66,00,00,00,06
1088 M9 DATA 60,00,00,60,04,B9,00,00,00,01
1089 w2 DATA 00,00,3A,92,41,F9,00,00,3C,3D
1090 eI DATA 22,39,00,00,3A,92,82,00,00,64
1091 hZ DATA 61,00,00,00,62,FC,00,0A,61,00
1092 lK DATA 00,B8,61,00,00,B4,60,00,00,3A
1093 8Q DATA 00,B9,00,00,00,08,00,00,3A,92
1094 7d DATA 66,00,00,06,60,00,00,2C,06,B9
1095 lH DATA 00,00,00,01,00,00,3A,92,41,F9
1096 0e DATA 00,00,30,3D,22,39,00,00,3A,92
1097 lF DATA 82,FC,00,64,61,00,00,80,82,FC
1098 gP DATA 00,0A,61,00,00,78,61,00,00,74
1099 lK DATA 4B,F9,00,00,3C,3D,2A,3C,00,00
1100 lZ DATA 00,03,2C,3C,00,00,00,63,2E,3C
1101 h0 DATA 00,00,3A,66,42,80,00,00,04,18
1102 aI DATA 08,39,00,06,00,BF,E0,01,67,FE
1103 nZ DATA 60,00,F4,BC,41,F9,00,00,3B,74
1104 2p DATA 42,81,32,39,00,00,3A,76,82,FC
1105 ky DATA 00,64,61,00,00,32,82,FC,00,0A
1106 tB DATA 61,00,00,2A,61,00,00,26,48,F9
1107 Nm DATA 00,00,3B,74,3A,3C,00,03,3C,30
1108 Qg DATA 01,03,3E,3C,00,02,23,F9,00,00
1109 Se DATA 3A,66,00,00,3A,6E,4E,B9,00,00
1110 UQ DATA 04,18,4E,75,D2,7C,00,30,10,C1
1111 lI DATA 42,41,48,41,4E,75,42,85,20,79
1112 aJ DATA 00,00,3B,10,4E,B9,00,00,2A,CE
1113 g0 DATA E3,5D,20,79,00,00,3B,18,4E,B9
1114 UF DATA 00,00,2A,CE,E3,5D,20,79,00,00
1115 92 DATA 3B,14,4E,B9,00,00,2A,CE,E3,5D
1116 zJ DATA 20,79,00,00,3B,10,4E,B9,00,00
1117 Qq DATA 2A,CE,4E,75,26,00,28,01,86,FC
1118 6S DATA 00,08,0C,FC,00,00,04,D8,43,48,43
1119 ee DATA 14,3C,00,07,94,03,05,30,40,00
1120 DZ DATA 67,00,00,06,08,0C,00,00,4E,75
1121 eh DATA 20,79,00,00,3A,4A,20,39,00,00
1122 80 DATA 3A,78,90,BC,00,00,00,01,D1,FC
1123 OT DATA 00,00,02,00,22,3C,00,00,00,1F
1124 ON DATA 20,FC,FF,FF,FF,FF,51,C9,FF,FF
1125 wS DATA 51,C8,FF,88,4E,75,20,79,00,00
1126 hV DATA 3A,4A,22,48,24,48,26,48,28,48

1127 Em DATA D3,FC,00,00,00,80,D5,FC,00,00
1128 Ff DATA 01,00,D7,FC,00,00,01,80,D9,FC
1129 Wn DATA 00,00,02,00,00,39,00,00,3A,78
1130 Kk DATA 90,BC,00,00,00,01,22,3C,00,00
1131 xm DATA 00,1F,42,87,24,18,8E,82,24,19
1132 pY DATA 8E,82,24,1A,8E,82,24,1B,8E,82
1133 4b DATA 2B,C7,51,C9,FF,EA,D1,FC,00,00
1134 XP DATA 02,00,03,FC,00,00,02,00,D5,FC
1135 aP DATA 00,00,02,00,D7,FC,00,00,02,00
1136 Lx DATA D9,FC,00,00,02,00,51,C8,FF,C2
1137 Dw DATA 4E,75,4E,B9,00,00,01,10,4E,B9
1138 Om DATA 00,00,00,06,4E,B9,00,00,00,66
1139 Wp DATA 4E,B9,00,00,2C,9C,4E,B9,00,00
1140 52 DATA 01,54,4E,B9,00,00,24,40,2C,78
1141 He DATA 00,04,22,3C,00,01,00,93,20,39
1142 vo DATA 00,00,3A,78,00,FC,02,80,4E,AE
1143 gG DATA FF,3A,23,00,00,00,3A,4A,67,DA
1144 aV DATA 23,00,00,00,3B,10,D0,BC,00,00
1145 Ag DATA 00,80,23,00,00,00,3E,14,D0,BC
1146 pC DATA 4E,B9,00,00,80,23,00,00,3B,18
1147 4z DATA D0,BC,00,00,00,80,23,00,00,00
1148 aQ DATA 3B,1C,D0,BC,00,00,00,80,23,00
1149 AP DATA 00,00,3B,20,33,FC,00,01,00,00
1150 N1 DATA 3A,76,4E,B9,00,00,2A,FE,08,39
1151 EJ DATA 00,06,00,FF,00,01,67,00,00,10
1152 MM DATA 08,39,00,0A,00,DF,00,16,67,00
1153 Gu DATA 00,DE,66,EE,22,79,00,90,3A,52
1154 RF DATA 42,85,42,86,3A,29,00,0E,1C,29
1155 D5 DATA 00,0C,BA,7C,00,00,6C,00,DA,86
1156 92 DATA BC,7C,00,00,6C,00,00,D8,84,61,00
1157 nq DATA D9,E0,60,EE,2C,78,00,04,20,39
1158 j7 DATA 00,00,3A,78,00,FC,02,80,22,79
1159 ez DATA 00,00,3A,4A,4E,AE,FF,2E,2C,78
1160 J6 DATA 00,04,20,3C,00,00,08,00,22,79
1161 x0 DATA 00,00,3A,5E,4E,AE,FF,2E,4E,B9
1162 G2 DATA 00,00,00,FO,4E,B9,00,00,00,42
1163 pk DATA 4E,F9,00,00,01,3C,2C,78,00,04
1164 Nt DATA 20,3C,00,00,08,00,22,3C,00,01
1165 RK DATA 00,03,4E,AE,FF,3A,23,00,00,00
1166 QH DATA 3A,5E,22,40,20,7C,00,00,36,3E
1167 JN DATA 20,3C,00,00,01,FF,22,D8,51,C8
1168 Jp DATA FF,FC,22,39,00,00,3A,5E,4E,F9
1169 7p DATA 00,00,3A,36,D1,FC,00,00,00,0A
1170 Xg DATA 20,3C,00,00,00,19,D3,90,D1,FC
1171 nF DATA 00,00,00,14,51,C8,FF,6E,4E,75
1172 aL DATA 22,79,00,00,3A,52,3A,29,00,0E
1173 mM DATA 3C,29,00,00,BA,7C,00,00,64,00
1174 gN DATA 00,2E,BC,7C,00,00,64,00,00,26
1175 5t DATA 33,F9,00,00,3A,72,00,00,3A,80
1176 n0 DATA 33,FC,00,00,00,00,3A,72,61,00
1177 x5 DATA D9,18,33,F9,00,00,3A,80,00,00
1178 LK DATA 3A,72,60,00,FE,EC,3A,7C,01,3F
1179 hH DATA 66,00,FE,EC,3C,7C,00,00,66,00
1180 Yh DATA FE,00,60,00,FF,1A,10,00,00,00
1181 xR DATA 01,40,01,00,00,00,00,01,30,02
1182 02 DATA 00,0F,00,00,00,00,00,00,00,00
1183 J1 DATA 00,00,00,00,00,00,00,00,00,00
1184 hK DATA 00,00,01,40,01,00,00,01,30,00
1185 7f DATA 00,00,00,01,18,80,00,00,00,00
1186 80 DATA 00,00,00,00,00,00,00,00,00,00
1187 Tk DATA 00,00,00,00,00,00,01,40,01,00
1188 oY DATA 01,40,01,00,00,00,01,00,01,00
1189 aB DATA 00,DF,00,DF,02,03,00,00,00,00
1190 rC DATA 00,01,18,00,00,00,00,00,00,00
1191 DT DATA 00,00,00,00,00,00,00,00,00,00
1192 Og DATA 00,00,00,00,00,FF,00,DF,00,DF
1193 Q1 DATA 00,DF,00,DF,00,00,00,5E,00,90
1194 f1 DATA 00,2D,02,03,00,00,00,00,00,01
1195 Jv DATA 18,00,00,00,30,46,00,00,00,00
1196 IY DATA 00,00,00,00,00,00,00,00,00,00
1197 Ov DATA 00,00,00,90,00,2D,00,90,00,2D
1198 yg DATA 00,0F,00,00,2E,76,60,00,00,57
1199 DL DATA 00,20,00,0A,00,00,08,01,00,04
1200 4b DATA 00,00,2E,22,00,00,00,00,00,00
1201 ze DATA 00,00,00,00,00,00,00,00,2E,46
1202 Lg DATA 00,01,00,00,00,00,00,00,00,00
1203 BK DATA 03,00,00,05,00,00,FE,32,00,00
1204 gd DATA 00,00,FF,FE,FF,FE,00,20,FF,FE
1205 Lx DATA 00,20,00,09,FF,FE,00,09,FF,FE
1206 CE DATA FF,FE,00,00,2E,6A,00,00,2E,70
1207 bv DATA 00,00,00,04,00,00,00,00,00,00
1208 Uk DATA 00,00,00,00,00,00,00,00,00,00
1209 z6 DATA 00,00,00,00,00,00,00,00,30,30
1210 Ih DATA 31,00,00,00,30,30,31,00,00,00

1211 Bv DATA 00,00,00,00,00,27,00,75,00,90
1212 Vh DATA 00,0A,00,00,00,01,00,04,00,00
1213 Dz DATA 2E,A2,00,00,00,00,00,00,00,00
1214 Po DATA 00,00,00,00,00,00,2E,C6,00,02
1215 h0 DATA 00,00,00,00,00,00,00,00,03,00
1216 bE DATA 00,05,00,00,2E,B2,00,00,00,00
1217 hD DATA FF,FE,FF,FE,00,90,FF,FE,00,90
1218 WE DATA 00,09,FF,FE,00,09,FF,FE,FF,FE
1219 IX DATA 00,00,2E,EA,00,00,2F,14,00,90
1220 c0 DATA 00,28,00,00,00,00,00,00,00,00
1221 hx DATA 00,00,00,00,00,00,00,00,00,00
1222 Vb DATA 00,00,00,00,00,00,4F,62,6A,65
1223 Zh DATA 6E,74,65,2E,41,4F,45,00,00,00
1224 k0 DATA 00,00,00,00,00,00,00,00,00,00
1225 11 DATA 00,00,00,00,00,00,00,00,00,00
1226 Qz DATA 00,00,00,00,00,00,00,00,4F,62
1227 Jg DATA 6A,65,6B,74,65,2E,41,4F,45,00
1228 c4 DATA 00,00,00,00,00,00,00,00,00,00
1229 p5 DATA 00,00,00,00,00,00,00,00,00,00
1230 q6 DATA 00,00,00,00,00,00,00,00,00,00
1231 eQ DATA 00,00,2F,9A,00,38,00,57,00,20
1232 Tq DATA 00,0A,00,00,08,01,00,04,00,00
1233 Kr DATA 2E,22,00,00,00,00,00,00,00,00
1234 fv DATA 00,00,00,00,00,00,2F,6A,00,01
1235 x1 DATA 00,00,00,00,00,00,2F,8E,00,00
1236 Ny DATA 2F,9A,00,00,00,04,00,00,00,00
1237 xD DATA 00,00,00,00,00,00,00,00,00,00
1238 yE DATA 00,00,00,00,00,00,00,00,00,00
1239 Fa DATA 30,30,31,00,00,00,30,30,31,00
1240 nF DATA 00,00,00,00,2F,66,00,88,00,57
1241 c1 DATA 00,20,00,0A,00,00,08,01,00,04
1242 kF DATA 00,00,2E,22,00,00,00,00,00,00
1243 fY DATA 00,00,00,00,00,00,00,00,2F,C6
1244 Sv DATA 00,04,00,00,00,00,00,00,2F,EA
1245 zH DATA 00,00,2F,FO,00,00,00,04,00,00
1246 eM DATA 00,00,00,00,00,00,00,00,00,00
1247 7N DATA 00,00,00,00,00,00,00,00,00,00
1248 KM DATA 00,00,30,30,31,00,00,00,30,30
1249 Qm DATA 31,00,00,00,00,00,00,00,00,20
1250 zk DATA 00,75,00,00,90,0A,00,00,00,01
1251 21 DATA 00,04,00,00,2E,A2,00,00,00,00
1252 OS DATA 00,00,00,00,00,00,00,00,00,00
1253 p7 DATA 30,22,00,05,00,00,00,00,00,00
1254 Y3 DATA 2E,EA,00,00,2F,14,00,00,00,28
1255 FV DATA 00,00,00,00,00,00,00,00,00,00
1256 cW DATA 00,00,00,00,00,00,00,00,00,00
1257 de DATA 00,00,00,00,00,00,00,00,00,39
1258 mu DATA 00,0F,00,20,00,0A,00,00,08,01
1259 g7 DATA 00,04,00,00,2E,22,00,00,00,00
1260 Ae DATA 00,00,00,00,00,00,00,00,00,00
1261 EX DATA 30,72,00,01,00,00,00,00,00,00
1262 U7 DATA 30,96,00,00,30,9C,00,00,00,04
1263 Nd DATA 00,00,00,00,00,00,00,00,00,00
1264 6w DATA 00,00,00,00,00,00,00,00,00,64
1265 1R DATA 00,00,00,00,31,30,30,00,00,00
1266 MH DATA 31,30,30,00,00,00,00,00,30,FE
1267 Ex DATA 00,38,00,61,00,20,0A,00,00,00
1268 We DATA 08,01,00,04,00,00,2E,22,00,00
1269 TJ DATA 00,00,00,00,00,00,00,00,00,00
1270 cD DATA 00,00,30,C8,00,01,00,00,00,00
1271 f7 DATA 00,00,30,F2,00,00,30,F8,00,00
1272 Ju DATA 00,04,00,00,00,00,00,00,00,00
1273 Kn DATA 00,00,00,00,00,00,00,00,00,00
1274 HT DATA 00,01,00,00,00,00,30,30,31,00
1275 HS DATA 00,00,30,30,31,00,00,00,00,00
1276 ke DATA 00,00,00,88,00,61,00,20,00,0A
1277 VF DATA 00,00,08,01,00,04,00,00,2F,22
1278 es DATA 00,00,00,00,00,00,00,00,00,00
1279 Jk DATA 00,00,00,00,31,2A,00,04,00,00
1280 2J DATA 00,00,00,00,31,4E,00,00,31,54
1281 n7 DATA 00,00,00,04,00,00,00,00,00,00
1282 gv DATA 00,00,00,00,00,00,00,00,00,00
1283 PG DATA 00,00,00,02,00,00,00,00,30,30
1284 de DATA 32,00,00,00,30,30,32,00,00,00
1285 Bq DATA 00,00,31,86,00,38,00,57,00,20
1286 L1 DATA 00,0A,00,00,08,01,00,04,00,00
1287 CJ DATA 2E,22,00,00,00,00,00,00,00,00
1288 yS DATA 00,00,00,00,00,00,31,86,00,01
1289 Jq DATA 00,00,00,00,00,00,31,AA,00,00
1290 bH DATA 31,B0,00,00,00,04,00,00,00,00
1291 p5 DATA 00,00,00,00,00,00,00,00,00,00
1292 tA DATA 00,00,00,00,00,01,00,00,00,00
1293 7S DATA 30,30,31,00,00,00,30,30,31,00
1294 9z DATA 00,00,00,00,32,12,00,88,00,57

1295 1t DATA 00,20,00,0A,00,00,08,01,00,04
1296 c9 DATA 00,00,2E,22,00,00,00,00,00,00
1297 dD DATA 00,00,00,00,00,00,00,00,31,EE
1298 46 DATA 00,04,00,00,00,00,00,00,32,06
1299 60 DATA 00,00,32,0C,00,00,00,04,00,00
1300 7E DATA 00,00,00,00,00,00,00,00,00,00
1301 7P DATA 00,00,00,00,00,00,00,02,00,00
1302 hA DATA 00,00,30,30,32,00,00,00,30,30
1303 cX DATA 32,00,00,00,00,00,00,00,02,00,7C
1304 c6 DATA 00,6B,00,20,00,0A,00,00,08,01
1305 qR DATA 00,04,00,00,2E,22,00,00,00,00
1306 4K DATA 00,00,00,00,00,00,00,00,00,00
1307 gm DATA 32,FE,00,04,00,00,00,00,00,00
1308 EL DATA 32,62,00,00,32,68,00,00,00,04
1309 7N DATA 00,00,00,00,00,00,00,00,00,00
1310 JH DATA 00,00,00,00,00,00,00,00,00,01
1311 m7 DATA 00,00,00,00,30,30,31,00,00,00
1312 cG DATA 30,30,31,00,00,00,00,00,05,55
1313 5U DATA 09,99,0B,EB,0E,EE,00,00,00,00
1314 CS DATA 00,00,00,00,00,00,00,00,00,00
1315 VT DATA 00,00,00,00,00,00,00,00,00,00
1316 5C DATA 00,07,00,0F,07,7F,07,07,07,07
1317 Ab DATA 0F,7F,03,33,0A,AA,0F,FF,07,00
1318 9p DATA 0F,00,0F,77,00,70,00,70,07,FF
1319 K2 DATA 00,02,00,28,00,0C,00,FF,00,10
1320 JV DATA 00,28,00,1A,00,FF,00,1E,00,EE
1321 pG DATA 00,28,00,FF,00,2C,00,28,00,36
1322 JY DATA 00,FF,00,3A,00,28,00,44,00,FF
1323 NJ DATA 00,48,00,28,00,52,00,FF,00,56
1324 Jt DATA 00,28,00,60,00,FF,00,64,00,EE
1325 7r DATA 00,6E,00,FF,00,72,00,28,00,7C
1326 KG DATA 00,FF,00,80,00,EE,00,8A,00,FF
1327 k1 DATA 00,8E,00,EE,00,98,00,FF,00,9C
1328 eK DATA 00,EE,00,A6,00,FF,00,AA,00,EE
1329 fm DATA 00,B4,00,FF,00,B8,00,EE,00,C2
1330 8N DATA 00,FF,00,C6,00,EE,00,00,00,FF
1331 LA DATA 00,04,00,28,00,1E,00,FF,01,09
1332 2L DATA 00,50,01,18,00,00,FF,01,09,00,2E
1333 JX DATA 01,16,00,3D,01,1A,00,3F,01,29
1334 VS DATA 00,4E,00,FF,00,3F,01,07,00,4E
1335 Ky DATA 01,09,00,3F,01,18,00,4E,01,00
1336 vd DATA 00,66,01,0F,00,75,01,11,00,66
1337 Tq DATA 01,20,00,75,01,00,00,77,01,0F
1338 19 DATA 00,86,01,11,00,77,01,20,00,86
1339 tS DATA 01,00,00,88,01,0F,00,97,01,11
1340 KZ DATA 00,88,01,20,00,97,01,00,00,99
1341 JK DATA 01,0F,00,A8,01,11,00,99,01,20
1342 d1 DATA 00,A8,00,FF,00,BC,00,FF,00,C3
1343 nJ DATA 01,20,00,BC,01,27,00,C3,00,FF
1344 e2 DATA 00,E2,01,02,00,E9,00,FF,00,FE
1345 4r DATA 01,02,00,FD,01,0B,00,E2,01,12
1346 EP DATA 00,E9,01,0B,00,F6,01,12,00,FD
1347 5m DATA 01,1B,00,E2,01,22,00,E9,01,1B
1348 xK DATA 00,F6,01,22,00,FD,01,01,00,BC
1349 Oq DATA 01,25,00,C3,00,00,07,42,00,00
1350 aS DATA 07,A0,00,00,07,FA,00,00,08,48
1351 db DATA 00,00,08,96,00,00,0A,28,00,00
1352 mI DATA 14,6E,00,00,14,A4,00,00,15,38
1353 tS DATA 00,00,15,84,00,00,16,02,00,00
1354 x0 DATA 19,A0,00,00,1A,FA,00,00,20,46
1355 v7 DATA 00,00,20,A2,00,00,21,00,00,00
1356 BG DATA 21,3C,00,00,21,78,00,00,21,BA
1357 96 DATA 00,00,21,FO,00,00,22,28,00,00
1358 f7 DATA 22,60,00,00,00,00,10,00,10
1359 PL DATA 00,02,00,00,00,00,03,00,00,00
1360 Au DATA 34,4A,00,00,FF,DE,00,10,00,10
1361 2Z DATA 00,02,00,00,00,40,03,00,00,00
1362 5M DATA 34,5E,00,11,FF,EF,00,10,00,10
1363 Jn DATA 00,02,00,00,00,80,03,00,00,00
1364 Kp DATA 34,72,FF,2F,FF,FF,00,10,00,10
1365 7N DATA 00,02,00,00,00,00,03,00,00,00
1366 b9 DATA 34,86,00,00,FF,EF,00,10,00,10
1367 e2 DATA 00,02,00,00,01,00,03,00,00,00
1368 BT DATA 00,00,00,00,00,00,10,00,10
1369 hE DATA 00,02,00,00,00,40,03,00,00,00
1370 JS DATA 34,AE,00,00,00,00,10,00,10
1371 bX DATA 00,02,00,00,00,00,03,00,00,00
1372 27 DATA 34,C2,FF,FF,FF,FF,00,10,00,10
1373 tx DATA 00,02,00,00,00,80,03,00,00,00
1374 pM DATA 34,D6,00,11,FF,FF,00,10,00,10
1375 HR DATA 00,02,00,00,00,00,03,00,00,00
1376 sv DATA 34,EA,00,00,FF,EF,00,10,00,10

Listing 1. (Fortsetzung)

1377 JV DATA 00,02,00,00,01,40,03,00,00,00
1378 LD DATA 00,00,00,00,00,00,00,10,00,10
1379 JN DATA 00,02,00,00,01,80,03,00,00,00
1380 NY DATA 35,12,00,11,00,00,00,10,00,10
1381 SI DATA 00,02,00,00,01,00,03,00,00,00
1382 ZF DATA 35,26,00,00,00,11,00,10,00,10
1383 XV DATA 00,02,00,00,02,00,03,00,00,00
1384 SE DATA 35,3A,00,11,00,11,00,10,00,10
1385 79 DATA 00,02,00,00,02,40,03,00,00,00
1386 KW DATA 35,4E,00,00,00,22,00,10,00,10
1387 HN DATA 00,02,00,00,02,80,03,00,00,00
1388 OD DATA 35,62,00,11,00,11,00,10,00,10
1389 GW DATA 00,02,00,00,02,00,03,00,00,00
1390 FV DATA 35,76,00,01,01,03,00,10,00,10
1391 A9 DATA 00,02,00,00,03,00,03,00,00,00
1392 Z4 DATA 35,8A,00,11,00,03,00,10,00,10
1393 KN DATA 00,02,00,00,03,00,03,00,00,00
1394 EA DATA 00,02,00,00,03,00,03,00,00,00
1395 JB DATA 00,02,00,00,03,00,03,00,00,00
1396 7V DATA 35,82,00,28,01,00,00,00,00,00
1397 04 DATA 00,02,00,00,03,00,03,00,00,00
1398 LE DATA 00,00,00,00,00,00,00,00,00,00
1399 JC DATA 00,02,00,00,03,00,03,00,00,00
1400 ZW DATA 35,DA,00,00,00,14,00,00,00,00
1401 0K DATA 00,02,00,00,03,00,03,00,00,00
1402 WU DATA 35,EE,00,10,00,00,00,00,00,00
1403 XG DATA 00,02,00,00,03,00,03,00,00,00
1404 IG DATA 36,02,00,10,00,14,00,00,00,00
1405 40 DATA 00,02,00,00,03,00,03,00,00,00
1406 5F DATA 36,16,00,20,00,00,00,00,00,00
1407 2K DATA 00,02,00,00,03,00,03,00,00,00
1408 JK DATA 36,2A,00,20,00,00,00,00,00,00
1409 BS DATA 00,02,00,00,03,00,03,00,00,00
1410 TO DATA 00,00,00,00,07,0E,0E,0E,0E,0E
1411 NX DATA 7E,FE,7E,FE,7E,FE,7E,FE,7E,FE
1412 EQ DATA 5F,FE,6F,FE,7F,FE,7E,FE,7D,FE
1413 05 DATA 7E,FE,00,00,00,01,00,21,00,21
1414 M2 DATA 00,21,00,21,00,21,00,21,00,21
1415 3X DATA 00,21,00,05,00,09,00,11,00,21
1416 bt DATA 00,41,00,81,7F,FF,00,00,7F,FE
1417 FF DATA 7F,8E,7F,DE,7F,EE,7F,FE,7F,FA
1418 SI DATA 7F,DE,7F,DE,7F,DE,7F,DE,7F,DE
1419 JW DATA 7F,DE,7F,DE,7F,DE,00,00,00,01
1420 90 DATA 02,02,02,01,04,01,08,01,10,01
1421 PD DATA 20,01,04,01,04,01,04,01,04,01
1422 v1 DATA 04,01,04,01,04,01,04,01,7F,FF
1423 5e DATA 00,00,7E,FE,7F,FE,7F,FE,7F,FE
1424 KU DATA 7F,FE,7F,FE,7F,FE,7F,FE,7F,FA
1425 a0 DATA 00,76,7F,EE,7F,DE,7F,FE,7F,FE
1426 Yy DATA 00,00,00,01,00,01,00,41,00,21
1427 Xz DATA 00,11,7F,89,00,05,00,03,00,01
1428 MQ DATA 00,01,00,00,00,01,00,01,00,01
1429 Iq DATA 00,01,7F,FF,00,00,7F,FE,7D,FE
1430 1A DATA 7E,FE,7F,FE,6E,00,5F,FE,3F,FE
1431 UJ DATA 7F,FE,7F,FE,7F,FE,7F,FE,7F,FE
1432 03 DATA 7F,FE,7F,FE,00,00,01,00,01
1433 hh DATA 00,01,00,01,00,01,00,01,00,01
1434 d2 DATA 00,01,40,01,20,01,1F,FF,08,01
1435 Fx DATA 04,01,02,01,00,01,7F,FF,00,00
1436 MX DATA 7F,8E,7F,8E,7F,DE,7F,8E,6F,7E
1437 AL DATA 17,FE,FB,FE,7F,DE,7E,8E,7E,FE
1438 8r DATA 7D,FE,7E,FE,7D,FE,7D,FE,00,00
1439 4H DATA 00,01,02,01,02,01,04,01,02,01
1440 XG DATA 01,09,00,17,00,21,04,01,68,01
1441 9S DATA 10,81,00,41,00,21,00,41,00,41
1442 Dg DATA 7F,FF,00,00,7F,FE,7F,FE,7F,FE
1443 ba DATA 7F,8E,6F,8E,5F,FE,3F,FE,7F,FE
1444 f3 DATA 7F,CA,7D,FE,7D,FE,7E,FE,7D,FE
1445 IR DATA 7E,FE,00,00,00,01,01,01,02,01
1446 9r DATA 04,01,02,01,02,00,00,00,03,00
1447 4T DATA 40,01,2C,01,10,41,00,41,00,21
1448 0a DATA 00,41,00,81,7F,FF,00,00,7F,FE
1449 hD DATA 47,CE,5F,FE,5F,FA,5F,FA,5F,FA
1450 gw DATA 5F,FA,5F,FA,5F,FA,5F,FA,5F,FA
1451 LZ DATA 5F,FA,40,02,7F,FE,00,00,00,01
1452 JJ DATA 00,01,07,01,07,49,07,45,07,05
1453 J3 DATA 00,05,00,05,00,05,00,05,00,05
1454 Gx DATA 00,05,00,05,1F,ED,00,01,7F,FF
1455 Wb DATA 00,00,7F,FE,03,86,3F,FE,3B,86
1456 7q DATA 3B,86,3B,86,3B,86,3B,86,3B,86
1457 Qo DATA 3B,8E,22,36,7F,FE,7F,FE,7F,FE
1458 Yc DATA 00,00,00,01,00,01,00,01,3E,39
1459 IO DATA 22,25,22,25,22,25,22,25,22,39
1460 cW DATA 22,29,22,29,22,25,3F,35,00,01

1461 ZN DATA 00,01,7F,FF,00,00,7E,FE,7E,FE
1462 C3 DATA 7E,FE,7E,FE,68,96,5E,FA,3E,FC
1463 V1 DATA 7E,FE,7E,FE,7E,FE,7E,FE,7E,FE
1464 7Q DATA 7E,FE,7E,FE,00,00,01,00,01
1465 Xn DATA 00,81,00,81,00,81,00,81,00,81
1466 Gp DATA 00,81,40,83,20,85,16,E9,08,91
1467 44 DATA 00,81,00,81,00,81,7F,FF,00,00
1468 ps DATA 7E,FE,7D,FE,7E,FE,7F,FE,7E,FE
1469 Ik DATA 7E,FE,00,02,7F,FE,7E,FE,7E,FE
1470 Ca DATA 7F,FE,7E,FE,7D,FE,7E,FE,00,00
1471 7y DATA 00,01,00,81,00,41,00,21,00,11
1472 Lr DATA 00,21,00,21,00,01,3F,FF,00,21
1473 72 DATA 00,21,00,11,00,21,00,41,00,81
1474 zp DATA 7F,FF,00,00,7F,FE,7F,FE,6F,FE
1475 PL DATA 5F,FE,6F,FE,6F,FE,6F,FE,6F,FE
1476 31 DATA 6F,FE,6F,FE,6F,FE,6F,FE,60,02
1477 55 DATA 7F,FE,00,00,00,01,00,01,04,01
1478 KZ DATA 02,01,01,01,02,01,02,01,02,01
1479 r1 DATA 02,01,02,01,03,FD,00,05,00,05
1480 7c DATA 00,01,00,01,7F,FF,00,00,7F,FE
1481 0x DATA 40,0E,5F,EE,5F,FE,5F,FE,5F,FE
1482 G1 DATA 5F,EE,5F,EE,5F,EE,5F,EE,48,0E
1483 02 DATA 7F,FE,7F,FE,7F,FE,00,00,00,01
1484 3x DATA 00,01,00,01,00,01,0F,FD,08,05
1485 7e DATA 08,05,08,05,08,05,08,05,08,05
1486 7Y DATA 08,05,08,05,0F,FD,00,01,7F,FF
1487 40 DATA 00,00,7F,FE,7C,FE,79,FE,7E,FE
1488 40 DATA 7F,FE,7E,3E,7D,FE,7D,FE,7D,FE
1489 05 DATA 7D,FE,7D,FE,71,DE,7F,FE,7F,FE
1490 4A DATA 00,00,00,01,00,01,00,01,01,01
1491 aS DATA 03,EL,01,01,00,01,03,01,01,01
1492 Ed DATA 01,01,01,01,01,01,01,01,07,F1
1493 g1 DATA 00,01,7F,FF,00,00,7F,FE,63,FE
1494 FM DATA 5D,FE,5D,FE,5D,FE,63,FE,77,FE
1495 eS DATA 7D,3E,7F,FE,47,FE,47,FE,77,FE
1496 FL DATA 7F,FE,7F,FE,00,00,00,01,00,01
1497 6c DATA 00,01,00,31,00,49,00,49,00,31
1498 Ig DATA 00,21,00,11,00,09,00,05,00,05
1499 7B DATA 00,05,07,FD,00,01,7F,FF,00,00
1500 eS DATA 6E,00,52,00,3E,00,7E,00,7E,00
1501 7L DATA 7E,00,00,00,01,00,01,00,01,00
1502 4P DATA 01,00,41,00,20,00,11,00,7E,00
1503 yE DATA 00,00,7E,00,7E,00,7E,00,7C,00
1504 KD DATA 4A,00,76,00,00,00,01,00,09,00
1505 1Q DATA 35,00,03,00,01,00,01,00,01,00
1506 1p DATA 7F,00,00,00,76,00,7A,00,7C,00
1507 yT DATA 7A,00,7A,00,7E,00,00,00,01,00
1508 Cf DATA 11,00,21,00,41,00,21,00,21,00
1509 0f DATA 01,00,7F,00,00,00,7E,00,5E,00
1510 2r DATA 5E,00,3E,00,5E,00,6E,00,00,00
1511 XZ DATA 01,00,01,00,05,00,05,00,03,00
1512 80 DATA 05,00,09,00,7F,00,00,00,00,00
1513 Pf DATA 00,00,00,00,00,00,00,00,00,00
1514 Qg DATA 00,00,00,00,00,00,00,00,00,00
1515 Rh DATA 00,00,00,00,00,00,00,00,00,00
1516 3h DATA 00,00,00,00,00,00,00,00,00,00
1517 Yp DATA 00,00,00,00,00,00,00,00,00,01
1518 4c DATA 00,00,00,01,00,00,00,00,00,00
1519 7c DATA 00,00,00,00,00,00,00,00,00,00
1520 4n DATA 00,00,00,00,00,00,00,00,00,00
1521 62 DATA 00,00,00,05,00,00,00,00,00,00
1522 00 DATA 00,00,00,00,00,00,00,00,00,00
1523 Zp DATA 00,00,00,00,00,00,00,00,00,00
1524 aq DATA 00,00,00,00,00,00,00,00,00,00
1525 br DATA 00,00,00,00,00,00,00,00,00,00
1526 cs DATA 00,00,00,00,00,00,00,00,00,00
1527 dt DATA 00,00,00,00,00,00,00,00,00,00
1528 ev DATA 00,00,00,00,00,00,00,00,00,00
1529 f1 DATA 00,00,00,00,00,00,00,00,00,00
1530 gw DATA 00,00,00,00,00,00,00,00,00,00
1531 0x DATA 00,00,00,00,00,00,00,00,00,00
1532 2M DATA 00,00,00,00,00,00,00,00,00,04
1533 3c DATA 00,20,00,05,00,00,00,00,00,00
1534 x0 DATA 00,00,00,00,00,00,00,00,00,00
1535 31 DATA 00,00,00,00,00,00,00,00,00,00
1536 JK DATA 00,00,00,00,00,00,00,00,67,72
1537 P9 DATA 61,70,68,69,63,73,2E,6C,69,62
1538 TM DATA 72,61,72,79,00,64,6F,73,2E,6C
1539 3P DATA 69,62,72,61,72,79,00,69,6E,74
1540 dC DATA 75,69,74,69,6F,6E,2E,6C,69,62
1541 29 DATA 72,61,72,79,00,4F,62,6A,65,6B
1542 2S DATA 74,20,4E,72,2E,52,20,47,20,42
1543 cW DATA 30,20,30,20,30,20,30,31,4C
1544 zi DATA 61,64,65,6E,53,70,65,69,63,68

1545 V3 DATA 65,72,6E,40,61,64,65,6E,2D,6E
1546 xA DATA 61,63,68,20,4F,62,6A,65,6B,74
1547 xA DATA 44,61,74,65,69,6E,61,6D,65,46
1548 S DATA 65,72,74,69,67,53,70,65,69,63
1549 70 DATA 68,65,72,6E,20,76,6F,6E,20,4F
1550 2V DATA 62,6A,65,68,74,6E,61,63,68,41
1551 75 DATA 4D,49,47,41,20,4F,62,6A,65,43
1552 4Y DATA 54,45,44,49,54,4F,52,20,56,31
1553 4k DATA 2E,32,77,72,69,74,74,65,6E,20
1554 3e DATA 31,39,38,38,20,62,79,4D,49,43
1555 42 DATA 48,41,45,40,20,42,45,52,54,53
1556 43 DATA 43,48,49,20,4D,61,72,6B,74,20
1557 4C DATA 26,20,54,65,63,68,6E,69,6B,5A
1558 7c DATA 65,69,67,65,20,4F,62,6A,65,6B
1559 45 DATA 74,20,4E,72,2E,41,62,62,72,75
1560 12 DATA 63,68,41,6E,69,6D,61,74,69,6F
1561 71 DATA 6E,20,76,6F,6E,20,4F,62,6A,65
1562 6C DATA 68,74,47,65,73,63,68,77,69,6E
1563 4E DATA 64,69,67,68,65,69,74,30,30,35
1564 5b DATA 4D,61,78,2E,20,4F,62,6A,65,6B
1565 7b DATA 74,7A,61,68,6C,4B,6F,70,69,65
1566 4E DATA 72,65,6E,20,76,6F,6E,20,4F,62
1567 4 DATA 6A,65,6B,74,41,6E,7A,61,68,60
1568 11 DATA 00,00,03,2C,00,00,03,36,00,00
1569 4K DATA 00,00,00,00,00,02,00,00,00,00
1570 54 DATA 00,00,00,18,00,00,00,1E,00,00
1571 5Z DATA 00,2A,00,00,00,30,00,00,00,3C
1572 46 DATA 00,00,00,48,00,00,00,52,00,00
1573 47 DATA 00,5C,00,00,00,68,00,00,00,72
1574 00 DATA 00,00,00,7C,00,00,00,82,00,00
1575 4W DATA 00,88,00,00,00,8E,00,00,00,98
1576 7b DATA 00,00,00,AA,00,00,00,AA,00,00
1577 0b DATA 00,32,00,00,00,BA,00,00,00,C6
1578 0b DATA 00,00,00,00,00,00,00,00,00,00
1579 51 DATA 00,7E,00,00,00,E6,00,00,00,F2
1580 62 DATA 00,00,01,F8,00,00,01,02,00,00
1581 73 DATA 01,36,00,00,01,3E,00,00,01,48
1582 6C DATA 00,00,00,56,00,00,01,5C,00,00
1583 47 DATA 01,6A,00,00,01,6A,00,00,01,70
1584 4 DATA 00,00,01,7E,00,00,01,94,00,00
1585 1F DATA 01,A2,00,00,01,A2,00,00,01,BE
1586 2w DATA 00,00,01,D8,00,00,01,FC,00,00
1587 4L DATA 02,CA,00,00,02,28,00,00,02,46
1588 4N DATA 00,00,02,54,00,00,02,6C,00,00
1589 70 DATA 02,86,00,00,02,94,00,00,02,86
1590 0Y DATA 00,00,02,D0,00,00,02,D6,00,00
1591 0c DATA 02,E4,00,00,02,FE,00,00,03,0A
1592 80 DATA 00,00,03,1C,00,00,03,44,00,00
1593 2Q DATA 03,5E,00,00,03,64,00,00,03,6A
1594 0 DATA 00,00,03,70,00,00,03,82,00,00
1595 41 DATA 03,88,00,00,03,9A,00,00,03,A0
1596 00 DATA 00,00,03,82,00,00,03,88,00,00
1597 V7 DATA 03,CA,00,00,03,DC,00,00,03,E2
1598 64 DATA 00,00,03,74,00,00,03,FA,00,00
1599 3F DATA 04,0C,00,00,04,12,00,00,04,1A
1600 80 DATA 00,00,04,20,00,00,04,36,00,00
1601 4V DATA 04,42,00,00,04,50,00,00,04,6C
1602 7c DATA 00,00,04,7A,00,00,04,8A,00,00
1603 7c DATA 04,90,00,00,04,9A,00,00,04,80
1604 7c DATA 00,00,04,8E,00,00,04,00,00,00
1605 2K DATA 04,DE,00,00,05,04,00,00,05,22
1606 3X DATA 00,00,05,26,00,00,05,2C,00,00
1607 f1 DATA 05,32,00,00,05,3E,00,00,05,62
1608 47 DATA 00,00,05,68,00,00,05,74,00,00
1609 35 DATA 05,66,00,00,05,D2,00,00,05,E0
1610 35 DATA 00,00,05,EE,00,00,05,FC,00,00
1611 5J DATA 06,02,00,00,06,14,00,00,06,18
1612 71 DATA 00,00,06,1E,00,00,06,30,00,00
1613 75 DATA 06,7A,00,00,06,84,00,00,06,8E
1614 7F DATA 00,00,06,9C,00,00,06,A2,00,00
1615 0v DATA 06,80,00,00,06,8E,00,00,06,D8
1616 A3 DATA 00,00,06,FE,00,00,07,22,00,00
1617 4R DATA 07,36,00,00,07,3A,00,00,07,46
1618 3C DATA 00,00,07,50,00,00,07,5A,00,00
1619 4x DATA 07,64,00,00,07,6E,00,00,07,A4
1620 7Y DATA 00,00,07,AE,00,00,07,88,00,00
1621 e3 DATA 07,C2,00,00,07,CC,00,00,07,FE
1622 4c DATA 00,00,08,08,00,00,08,12,00,00
1623 7T DATA 08,1C,00,00,08,26,00,00,08,4C
1624 7c DATA 00,00,08,56,00,00,08,60,00,00
1625 4F DATA 08,6A,00,00,08,74,00,00,08,9A
1626 7c DATA 00,00,08,AA,00,00,08,AA,00,00
1627 A DATA 08,B0,00,00,08,00,00,08,D2
1628 7p DATA 00,00,08,DC,00,00,08,E6,00,00

1629 X2 DATA 08,00,00,00,09,24,00,00,09,2E
1630 eT DATA 00,00,09,38,00,00,09,42,00,00
1631 7N DATA 09,74,00,00,09,7E,00,00,09,88
1632 yd DATA 00,00,09,92,00,00,09,B6,00,00
1633 NL DATA 09,00,00,00,09,CA,00,00,09,34
1634 Sg DATA 00,00,09,FC,00,00,0A,02,00,00
1635 SM DATA 0A,08,00,00,0A,20,00,00,0A,2A
1636 FE DATA 00,00,0A,30,00,00,0A,3A,00,00
1637 5z DATA 0A,48,00,00,0A,56,00,00,0A,60
1638 Ky DATA 00,00,0A,6E,00,00,0A,88,00,00
1639 Ty DATA 0A,8C,00,00,0A,92,00,00,0A,A4
1640 Un DATA 00,00,0A,AA,00,00,0A,BC,00,00
1641 p9 DATA 0A,C2,00,00,0A,D4,00,00,0A,EF
1642 sS DATA 00,00,0B,08,00,00,0B,22,00,00
1643 1o DATA 0B,3C,00,00,0B,AA,00,00,0B,BO
1644 1x DATA 00,00,0B,BE,00,00,0B,C4,00,00
1645 PR DATA 0B,CE,00,00,0B,E0,00,00,0B,EA
1646 OR DATA 00,00,0C,00,00,00,0C,06,00,00
1647 k8 DATA 0C,0A,00,00,00,10,00,00,0C,1A
1648 Is DATA 00,00,0C,24,00,00,0C,2C,00,00
1649 MJ DATA 0C,3A,00,00,0C,44,00,00,0C,52
1650 eB DATA 00,00,0C,6C,00,00,0C,86,00,00
1651 5U DATA 0C,A0,00,00,0C,BA,00,00,0C,D4
1652 2Q DATA 00,00,0C,EE,00,00,0D,08,00,00
1653 2I DATA 0D,1A,00,00,0D,20,00,00,0D,32
1654 k4 DATA 00,00,0D,38,00,00,0D,4A,00,00
1655 US DATA 0D,64,00,00,0D,6A,00,00,0D,62
1656 IO DATA 00,00,0D,AC,00,00,0D,BC,00,00
1657 YJ DATA 0D,CA,00,00,0D,00,00,0D,06
1658 Fr DATA 00,00,0D,E4,00,00,0D,EA,00,00
1659 7E DATA 0D,F0,00,00,0D,F6,00,00,0D,04
1660 Pa DATA 00,00,0E,10,00,00,0E,1A,00,00
1661 e2 DATA 0E,48,00,00,0E,62,00,00,0E,6E
1662 mt DATA 00,00,0E,78,00,00,0E,8C,00,00
1663 1b DATA 0E,94,00,00,0E,A4,00,00,0E,B6
1664 Da DATA 00,00,0E,C4,00,00,0E,CA,00,00
1665 Tt DATA 0E,D0,00,00,0E,D6,00,00,0E,EA
1666 aQ DATA 00,00,0E,EA,00,00,0E,F0,00,00
1667 Jv DATA 0E,F6,00,00,0F,04,00,00,0F,0A
1668 pJ DATA 00,00,0F,14,00,00,0F,1F,00,00
1669 On DATA 0F,28,00,00,0F,2E,00,00,0F,48
1670 ri DATA 00,00,0F,6C,00,00,10,0C,00,00
1671 Pn DATA 10,3E,00,00,10,4E,00,00,10,84
1672 7V DATA 00,00,10,8A,00,00,10,94,00,00
1673 QZ DATA 10,A6,00,00,10,B0,00,00,10,C6
1674 YF DATA 00,00,10,CC,00,00,10,D0,00,00
1675 ev DATA 10,D6,00,00,10,E0,00,00,10,EA
1676 SI DATA 00,00,10,F2,00,00,11,00,00,00
1677 pE DATA 11,0A,00,00,11,18,00,00,11,32
1678 2W DATA 00,00,11,4C,00,00,11,66,00,00
1679 gr DATA 11,80,00,00,11,9A,00,00,11,B4
1680 RG DATA 00,00,11,CE,00,00,11,EB,00,00
1681 Eq DATA 11,FA,00,00,12,00,00,00,12,12
1682 8h DATA 00,00,12,18,00,00,12,2A,00,00
1683 eq DATA 12,30,00,00,12,42,00,00,12,5C
1684 hv DATA 00,00,12,62,00,00,12,7A,00,00
1685 ce DATA 12,A4,00,00,12,84,00,00,12,C2
1686 1l DATA 00,00,12,C8,00,00,12,CE,00,00
1687 xD DATA 12,DC,00,00,12,E2,00,00,12,ES
1688 Ub DATA 00,00,12,EE,00,00,13,00,00,00
1689 zH DATA 13,0E,00,00,13,18,00,00,13,26
1690 xH DATA 00,00,13,2C,00,00,13,32,00,00
1691 Bi DATA 13,3E,00,00,13,44,00,00,13,4A
1692 ju DATA 00,00,13,50,00,00,13,5A,00,00
1693 Eq DATA 13,66,00,00,13,7A,00,00,13,9A
1694 Sn DATA 00,00,13,AD,00,00,13,BO,00,00
1695 Wp DATA 13,CA,00,00,13,DO,00,00,13,24
1696 1P DATA 00,00,13,EA,00,00,13,FA,00,00
1697 3g DATA 14,00,00,00,14,06,00,00,14,16
1698 gl DATA 00,00,14,24,00,00,14,2A,00,00
1699 Ho DATA 14,38,00,00,14,3E,00,00,14,48
1700 JI DATA 00,00,14,4E,00,00,14,5C,00,00
1701 7p DATA 14,62,00,00,14,70,00,00,14,76
1702 rz DATA 00,00,14,86,00,00,14,9C,00,00
1703 NX DATA 14,A6,00,00,14,BO,00,00,14,BA
1704 ZI DATA 00,00,14,C4,00,00,15,3A,00,00
1705 KD DATA 15,44,00,00,15,4E,00,00,15,58
1706 ed DATA 00,00,15,86,00,00,15,90,00,00
1707 Qw DATA 15,9A,00,00,15,A4,00,00,16,04
1708 26 DATA 00,00,16,16,00,00,16,20,00,00
1709 pQ DATA 16,36,00,00,16,3C,00,00,16,40
1710 VQ DATA 00,00,16,46,00,00,16,50,00,00
1711 YI DATA 16,5A,00,00,16,62,00,00,16,70
1712 qv DATA 00,00,16,7A,00,00,16,88,00,00

1713 dJ DATA 16,A2,00,00,16,BC,00,00,16,D6
1714 4r DATA 00,00,16,F0,00,00,17,0A,00,00
1715 eY DATA 17,24,00,00,17,3E,00,00,17,58
1716 PM DATA 00,00,17,6A,00,00,17,70,00,00
1717 Sh DATA 17,82,00,00,17,88,00,00,17,9A
1718 en DATA 00,00,17,AD,00,00,17,B2,00,00
1719 eY DATA 17,CC,00,00,17,D2,00,00,17,EA
1720 UQ DATA 00,00,18,14,00,00,18,20,00,00
1721 qX DATA 18,26,00,00,18,34,00,00,18,3A
1722 Ay DATA 00,00,18,40,00,00,18,46,00,00
1723 OB DATA 18,54,00,00,18,62,00,00,18,68
1724 fd DATA 00,00,18,76,00,00,18,7C,00,00
1725 wU DATA 18,86,00,00,18,8C,00,00,18,98
1726 JY DATA 00,00,18,A2,00,00,18,AB,00,00
1727 uE DATA 18,B4,00,00,18,C4,00,00,18,CE
1728 Hn DATA 00,00,18,DE,00,00,18,EE,00,00
1729 qh DATA 18,F6,00,00,19,06,00,00,19,18
1730 kD DATA 00,00,19,1E,00,00,19,26,00,00
1731 UF DATA 19,36,00,00,19,3C,00,00,19,4C
1732 TQ DATA 00,00,19,54,00,00,19,5A,00,00
1733 OI DATA 19,6A,00,00,19,84,00,00,19,8E
1734 V9 DATA 00,00,19,94,00,00,19,A2,00,00
1735 BJ DATA 19,AB,00,00,19,B8,00,00,19,C2
1736 ax DATA 00,00,19,DO,00,00,19,DA,00,00
1737 He DATA 19,E8,00,00,19,FE,00,00,1A,02
1738 JC DATA 00,00,1A,0E,00,00,1A,1A,00,00
1739 BW DATA 1A,20,00,00,1A,32,00,00,1A,38
1740 7g DATA 00,00,1A,44,00,00,1A,5C,00,00
1741 6J DATA 1A,6E,00,00,1A,68,00,00,1A,7A
1742 JB DATA 00,00,1A,80,00,00,1A,92,00,00
1743 HL DATA 1A,AC,00,00,1A,BC,00,00,1A,B2
1744 kE DATA 00,00,1A,EB,00,00,1A,F6,00,00
1745 1b DATA 1A,FC,00,00,1B,00,00,1B,06
1746 R5 DATA 00,00,1B,10,00,00,1B,1A,00,00
1747 IF DATA 1B,22,00,00,1B,30,00,00,1B,3A
1748 Dr DATA 00,00,1B,48,00,00,1B,5A,00,00
1749 8f DATA 1B,64,00,00,1B,7A,00,00,1B,94
1750 17 DATA 00,00,1B,AE,00,00,1B,C8,00,00
1751 en DATA 1B,E2,00,00,1B,FC,00,00,1C,0E
1752 Af DATA 00,00,1C,14,00,00,1C,26,00,00
1753 cA DATA 1C,2C,00,00,1C,3E,00,00,1C,58
1754 4e DATA 00,00,1C,68,00,00,1C,7C,00,00
1755 U7 DATA 1C,94,00,00,1C,A2,00,00,1C,ED
1756 YS DATA 00,00,1C,B6,00,00,1C,BC,00,00
1757 1n DATA 1C,C6,00,00,1C,CA,00,00,1C,E4
1758 Eb DATA 00,00,1C,E2,00,00,1C,F8,00,00
1759 Pg DATA 1C,FE,00,00,1D,02,00,00,1D,0C
1760 Ko DATA 00,00,1D,10,00,00,1D,1A,00,00
1761 HY DATA 1D,1A,00,00,1D,1A,00,00,1D,20
1762 Ir DATA 00,00,1D,2A,00,00,1D,2A,00,00
1763 qQ DATA 1D,3A,00,00,1D,3A,00,00,1D,44
1764 V5 DATA 1D,4E,00,00,1D,5C,00,00,1D,70
1765 AO DATA 1D,6A,00,00,1D,6A,00,00,1D,74
1766 YN DATA 00,00,1D,8A,00,00,1D,8A,00,00
1767 S5 DATA 1D,96,00,00,1D,9A,00,00,1D,B6
1768 Ve DATA 00,00,1D,EE,00,00,1D,FF,00,00
1769 o9 DATA 1E,04,00,00,1E,06,00,00,1E,0C
1770 4J DATA 00,00,1E,1E,00,00,1E,1F,00,00
1771 1I DATA 1E,30,00,00,1E,3C,00,00,1E,3A
1772 5J DATA 00,00,1E,44,00,00,1E,52,00,00
1773 8E DATA 1E,60,00,00,1E,6A,00,00,1E,AC
1774 eV DATA 00,00,1E,B6,00,00,1E,C0,00,00
1775 Di DATA 1E,CA,00,00,1E,D4,00,00,1E,DE
1776 EA DATA 00,00,1E,E2,00,00,1E,FD,00,00
1777 74 DATA 1E,F6,00,00,1F,00,00,00,1F,1E
1778 sL DATA 00,00,1F,24,00,00,1F,2A,00,00
1779 kP DATA 1F,36,00,00,1F,66,00,00,1F,E6
1780 sX DATA 00,00,1F,72,00,00,1F,7C,00,00
1781 AH DATA 20,06,00,00,20,10,00,00,20,1A
1782 pD DATA 00,00,20,20,00,00,20,26,00,00
1783 ZJ DATA 20,30,00,00,20,34,00,00,20,3E
1784 Ht DATA 00,00,20,48,00,00,20,5E,00,00
1785 k5 DATA 20,68,00,00,20,72,00,00,20,7C
1786 m9 DATA 00,00,20,86,00,00,20,8E,00,00
1787 ch DATA 20,94,00,00,20,9A,00,00,20,A4
1788 Qn DATA 00,00,20,AA,00,00,20,BC,00,00
1789 Zn DATA 20,C6,00,00,20,30,00,00,20,DA
1790 XD DATA 00,00,20,E4,00,00,20,EC,00,00
1791 Vg DATA 20,F2,00,00,20,F8,00,00,21,04
1792 sS DATA 00,00,21,10,00,00,21,2E,00,00
1793 1U DATA 21,34,00,00,21,40,00,00,21,4C
1794 p6 DATA 00,00,21,6A,00,00,21,70,00,00
1795 8t DATA 21,7C,00,00,21,88,00,00,21,A6
1796 2w DATA 00,00,21,AC,00,00,21,B6,00,00

1797 EM DATA 21,C4,00,00,21,E2,00,00,21,E8
1798 md DATA 00,00,21,F4,00,00,22,00,00,00
1799 5i DATA 22,1A,00,00,22,20,00,00,22,2C
1800 4d DATA 00,00,22,38,00,00,22,52,00,00
1801 zT DATA 22,58,00,00,22,62,00,00,22,68
1802 3B DATA 00,00,22,72,00,00,22,7C,00,00
1803 aT DATA 22,84,00,00,22,92,00,00,22,9C
1804 v5 DATA 00,00,22,B2,00,00,22,C8,00,00
1805 9v DATA 22,D6,00,00,22,FA,00,00,23,14
1806 3m DATA 00,00,23,2C,00,00,23,46,00,00
1807 4p DATA 23,4C,00,00,23,5E,00,00,23,64
1808 7b DATA 00,00,23,76,00,00,23,90,00,00
1809 v1 DATA 23,96,00,00,23,AE,00,00,23,DB
1810 B1 DATA 00,00,23,24,00,00,23,EA,00,00
1811 ZE DATA 23,FB,00,00,24,06,00,00,24,0C
1812 SL DATA 00,00,24,12,00,00,24,1C,00,00
1813 1Z DATA 24,20,00,00,24,26,00,00,24,2C
1814 1J DATA 00,00,24,3C,00,00,24,42,00,00
1815 1A DATA 24,4E,00,00,24,5A,00,00,24,66
1816 nH DATA 00,00,24,72,00,00,24,78,00,00
1817 9e DATA 24,7E,00,00,24,84,00,00,24,92
1818 2c DATA 00,00,24,98,00,00,24,A6,00,00
1819 7B DATA 24,BA,00,00,24,BA,00,00,24,C0
1820 3a DATA 00,00,24,C6,00,00,24,DO,00,00
1821 rC DATA 24,DA,00,00,24,E2,00,00,24,FO
1822 73 DATA 00,00,24,FA,00,00,25,10,00,00
1823 10 DATA 25,26,00,00,25,34,00,00,25,38
1824 1m DATA 00,00,25,72,00,00,25,8A,00,00
1825 S3 DATA 25,A4,00,00,25,AA,00,00,25,BC
1826 4v DATA 00,00,25,C2,00,00,25,D4,00,00
1827 41 DATA 25,EE,00,00,25,FA,00,00,26,0C
1828 wY DATA 00,00,26,36,00,00,26,42,00,00
1829 7U DATA 26,48,00,00,26,56,00,00,26,64
1830 9s DATA 00,00,26,6A,00,00,26,6E,00,00
1831 PG DATA 26,74,00,00,26,7A,00,00,26,80
1832 XI DATA 00,00,26,92,00,00,26,A0,00,00
1833 0Y DATA 26,A6,00,00,26,C6,00,00,26,D4
1834 0Y DATA 00,00,26,DE,00,00,26,EE,00,00
1835 1g DATA 26,F4,00,00,27,00,00,00,27,0C
1836 1r DATA 00,00,27,18,00,00,27,24,00,00
1837 1a DATA 27,2C,00,00,27,32,00,00,27,36
1838 3S DATA 00,00,27,3C,00,00,27,42,00,00
1839 7D DATA 27,4C,00,00,27,50,00,00,27,5E
1840 1F DATA 00,00,27,6E,00,00,27,7C,00,00
1841 Vy DATA 27,9C,00,00,27,AB,00,00,27,AE
1842 4d DATA 00,00,27,84,00,00,27,D6,00,00
1843 3c DATA 27,DC,00,00,27,EC,00,00,28,06
1844 bx DATA 00,00,28,2A,00,00,28,38,00,00
1845 4A DATA 28,56,00,00,28,8C,00,00,28,AA
1846 82 DATA 00,00,28,CC,00,00,28,D6,00,00
1847 7M DATA 28,E0,00,00,28,EA,00,00,28,F4
1848 pf DATA 00,00,29,50,00,00,29,AD,00,00
1849 8y DATA 29,B2,00,00,29,88,00,00,29,BE
1850 7n DATA 00,00,29,E0,00,00,29,F2,00,00
1851 7n DATA 29,FE,00,00,29,FE,00,00,2A,18
1852 c3 DATA 00,00,2A,30,00,00,2A,44,00,00
1853 13 DATA 2A,4C,00,00,2A,66,00,00,2A,78
1854 18 DATA 00,00,2A,7C,00,00,2A,82,00,00
1855 Pw DATA 2A,98,00,00,2A,9E,00,00,2A,A6
1856 K9 DATA 00,00,2A,AC,00,00,2A,B4,00,00
1857 3A DATA 2A,BA,00,00,2A,C2,00,00,2A,C8
1858 1x DATA 00,00,2A,FA,00,00,2A,FA,00,00
1859 7e DATA 2B,22,00,00,2B,48,00,00,2B,96
1860 65 DATA 00,00,2B,9C,00,00,2B,A2,00,00
1861 vI DATA 2B,AB,00,00,2B,AE,00,00,2B,B4
1862 7u DATA 00,00,2B,C4,00,00,2B,D2,00,00
1863 9Q DATA 2B,DA,00,00,2B,E6,00,00,2B,F2
1864 Fo DATA 00,00,2B,FE,00,00,2C,0A,00,00
1865 4o DATA 2C,12,00,00,2C,18,00,00,2C,38
1866 11 DATA 00,00,2C,64,00,00,2C,6E,00,00
1867 43 DATA 2C,82,00,00,2C,8C,00,00,2C,92
1868 ar DATA 00,00,2C,98,00,00,2C,B2,00,00
1869 gx DATA 2C,BA,00,00,2C,C0,00,00,2C,D2
1870 2b DATA 00,00,2C,F2,00,00,2D,10,00,00
1871 fC DATA 2D,14,00,00,2D,1C,00,00,2D,26
1872 1r DATA 00,00,2D,2A,00,00,2D,D8,00,00
1873 7p DATA 2D,F6,00,00,2E,08,00,00,2E,18
1874 0c DATA 00,00,2E,2A,00,00,2E,46,00,00
1875 7Z DATA 2E,4A,00,00,2E,88,00,00,2E,98
1876 1T DATA 00,00,2E,AA,00,00,2E,C6,00,00
1877 mh DATA 2E,CA,00,00,2F,3E,00,00,2F,50
1878 4J DATA 00,00,2F,60,00,00,2F,6A,00,00

Listing 1. (Fortsetzung)


```

1879 VV DATA 2F,6E,00,00,2F,9A,00,00,2F,AC
1880 Vb DATA 00,00,2F,8C,00,00,2F,C6,00,00
1881 ED DATA 2F,CA,00,00,30,08,00,00,30,18
1882 Ue DATA 00,00,30,22,00,00,30,26,00,00
1883 KQ DATA 30,58,00,00,30,68,00,00,30,72
1884 bu DATA 00,00,30,76,00,00,30,A2,00,00
1885 YI DATA 30,B4,00,00,30,C4,00,00,30,CE
1886 xD DATA 00,00,30,D2,00,00,31,10,00,00
1887 3y DATA 31,20,00,00,31,2A,00,00,31,2E
1888 YD DATA 00,00,31,5A,00,00,31,6C,00,00
1889 3a DATA 31,7C,00,00,31,86,00,00,31,8A
1890 P5 DATA 00,00,31,B6,00,00,31,C8,00,00
1891 F4 DATA 31,D8,00,00,31,E2,00,00,31,E6
1892 xG DATA 00,00,32,24,00,00,32,34,00,00
1893 xg DATA 32,3E,00,00,32,42,00,00,33,DE
1894 vF DATA 00,00,33,E2,00,00,33,E6,00,00
1895 Fr DATA 33,EA,00,00,33,EE,00,00,33,F2

```

```

1896 wS DATA 00,00,33,F6,00,00,33,FA,00,00
1897 Ad DATA 33,FE,00,00,34,02,00,00,34,06
1898 YK DATA 00,00,34,0A,00,00,34,0E,00,00
1899 tX DATA 34,12,00,00,34,16,00,00,34,1A
1900 gB DATA 00,00,34,1E,00,00,34,22,00,00
1901 LI DATA 34,26,00,00,34,2A,00,00,34,2E
1902 eo DATA 00,00,34,32,00,00,34,46,00,00
1903 rn DATA 34,5A,00,00,34,6E,00,00,34,82
1904 AB DATA 00,00,34,AA,00,00,34,BE,00,00
1905 Sn DATA 34,D2,00,00,34,E6,00,00,35,0E
1906 Jt DATA 00,00,35,22,00,00,35,36,00,00
1907 uq DATA 35,4A,00,00,35,5E,00,00,35,72
1908 7Q DATA 00,00,35,86,00,00,35,AE,00,00
1909 ad DATA 35,D6,00,00,35,EA,00,00,35,FE
1910 oy DATA 00,00,36,12,00,00,36,26,00,00
1911 Su DATA 00,1D,00,00,00,01,00,00,0E,2C
1912 hE DATA 00,00,0E,3E,00,00,0E,54,00,00

```

```

1913 3m DATA 7E,68,0D,00,0E,AC,00,00,0F,40
1914 ag DATA 00,00,0F,54,00,00,0F,64,00,00
1915 7a DATA 0F,76,00,00,0F,82,00,00,0F,AC
1916 qu DATA 00,00,0A,08,00,00,0F,D0,00,00
1917 77 DATA 13,7E,00,00,13,86,00,00,13,AA
1918 75 DATA 06,00,13,B6,00,00,15,B6,00,00
1919 tQ DATA 15,00,00,1E,B0,00,00,1E,BA
1920 7C DATA 00,00,1E,C4,00,00,1E,CE,00,00
1921 p4 DATA 1E,D8,00,00,1F,EE,00,00,1F,FE
1922 71 DATA 00,00,2C,32,00,00,2D,00,00,00
1923 72 DATA 20,16,00,00,00,00,00,00,03,F2
1924 77 DATA 00,00,03,EB,00,00,01,00,00,00
1925 qK DATA 03,F2
(C) 1987 MBT

```

Listing 1. (Schluß)

Der Konvertierer

Von Andreas Lietz

Haben Sie schon einmal mit dem langsamen und umständlichen Amiga-Basic-Objekteditor gearbeitet? Dann werden Sie es sehr zu schätzen wissen, daß Sie jetzt die Objekte Ihrer Träume komfortabel mit dem »Amiga Object Editor« (AOE) aus dem vorangehenden Artikel (Seite 6).

»AOE-Convert« wandelt sie in das BOB-Format, auf das Sie mit Hilfe der OBJECT-Befehle des Amiga-Basic bequem Zugriff haben.

Da AOE-Convert in Assembler geschrieben ist, haben wir das Programm als Basic-Lader (Listing 1) abgedruckt. Assembler-Interessierte finden in Listing 2 den Quellcode des Programms.

Tippen Sie bitte Listing 1 mit Hilfe des Checksummers auf Seite 159 ab und speichern es als »AOE-Convert.bas«. Nach dem Start mit »RUN« erzeugt der Basic-Lader das lauffähige Programm »AOE-Convert« auf Diskette.

Dieses Programm rufen Sie durch Eingabe seines Namens aus dem CLI auf; Sie können ihm auch mit dem CLI-Befehl

```
copy icon.info
AOE-Convert.info
```

ein Icon zuweisen und das Programm über die Workbench aufrufen.

Die Anwendung des Programms ist ganz einfach: Nachdem Sie es gestartet haben, erscheint am oberen Bildschirmrand ein Fenster (Bild 1), in dem nach der AOE-Datei gefragt wird. Geben Sie also den Na-

Mit dem »Amiga Object Editor« aus dem Sonderheft 2 konnte man zwar wunderschöne Sprites entwerfen, aber sie ließen sich nicht mit Amiga-Basic verwenden. Mit »AOE-Convert« bauen Sie diese Objekte problemlos in Ihre Basic-Programme ein.

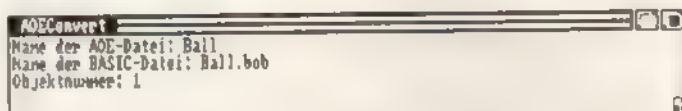


Bild 1. Das Fenster von »AOE-Convert«

men der Datei ein, die Sie konvertieren wollen.

Nun fragt das Programm nach der Datei, in die das konvertierte Objekt geschrieben werden soll.

Als letztes erfolgt die Frage nach der Objektnummer. Amiga-Basic kann nämlich nur einzelne BOBs in einer Datei verarbeiten – AOE speichert aber unter Umständen mehrere Objekte zusammen auf Diskette.

Wenn Sie auf diese Frage mit <RETURN> antworten, wird das erste Objekt der Datei konvertiert. Falls das Programm nach dieser Frage kommentarlos abbricht, ist ein Fehler aufgetreten.

Möglicherweise wurde die Quelldatei nicht gefunden oder sie war keine AOE-Datei. Auch beim Schreiben der Zieldatei kann etwas nicht funktionieren.

Nachdem das Objekt konvertiert wurde, fragt das Programm nach dem Namen eines weiteren Objekts. Ist alles konvertiert, verlassen Sie »AOE-Convert«, indem Sie <RETURN> ohne zusätzliche Eingabe drücken.

Achtung: Sie sollten nach dem Konvertieren die AOE-Datei nicht löschen! Vielleicht möchten Sie ja später noch einmal an dem Objekt arbeiten. Außerdem wird diese Datei für die richtige Einstellung der Farben im Basic-Programm benötigt.

Die konvertierten BOBs lassen sich ganz normal mit dem OBJECT.SHAPE-Befehl von Amiga-Basic laden. Sie werden in der richtigen Größe angezeigt, wenn Sie einen niedrig-auflösenden Screen (320 x 256 Punkte) mit 16 Farben verwenden. Damit das Objekt die richtigen Farben bekommt, müßte man mit Hilfe des Palette-Befehls

fehls nun mühsam die Farben einstellen, mit denen das Objekt zuvor erzeugt wurde.

Diese Arbeit nimmt Ihnen das kleine Programm in Listing 3 ab. Es macht sich die Tatsache zunutze, daß der Amiga Object Editor in seinen Dateien die Farben des Objektes mit speichert.

Bauen Sie Listing 3 an einer Stelle Ihres Programms ein, an der der Screen und das Fenster bereits geöffnet wurden, in denen sich Ihr Objekt bewegen soll. Wenn Sie in den OPEN-Befehl in Zeile 2 den Namen Ihrer AOE-Datei einsetzen, werden die Farben automatisch richtig eingestellt. Ein Beispiel, bei dem das Objekt auf dem Bildschirm bewegt wird, sehen Sie in Listing 4.

Natürlich müssen Sie auch hier in Zeile 4 und 15 die OPEN-Befehle entsprechend anpassen.

Damit haben Sie von Basic aus Zugriff auf die Dateien des »Amiga Object Editor« (AOE) aus dieser Ausgabe (Seite 6).

Sie stellen mit AOE einen BOB her, bringen das Ergebnis mit Hilfe von AOE-Convert auf Basic-Format und binden diese Datei als BOB in Ihre eigenen Programme ein. *kn*

Programmname:	AOE-Convert.bas
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	Basiclader für »AOE-Convert«
Programmautor: Michael Petersen	

1 Dim %DIM Generiert "lauffähiges Programm"	
2 ag %LF	
3 KQ OPEN "AOE-Convert", OF, CPU, AS 1	


```

4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 r1 IF wert1>64 THEN wert1=wert1-87
    ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7e wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87
    ELSE wert2=wert2-48
12 w1 wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 J8G NEXT
15 3n CLOSE 1
16 0w END
17 yw Werte.
18 9A DATA 1092
19 pn DATA 00,00,03,f3,00,00,00,00,00,00
20 na DATA 00,03,00,00,00,00,00,00,00,02
21 IL DATA 00,00,00,aa,00,00,00,19,00,00
22 x8 DATA 00,aa,00,00,03,e9,00,00,00,aa
23 1c DATA 2c,78,00,04,93,e9,4e,ae,fe,da
24 0H DATA 28,40,4e,ae,00,aa,66,16,41,ec
25 x3 DATA 00,5c,4e,ae,fe,80,41,ec,00,5c
26 r9 DATA 4e,ae,fe,8c,23,e0,00,00,00,00
27 23 DATA 43,f9,00,00,00,00,70,00,4e,ae
28 y1 DATA f0,d8,23,e0,00,00,00,04,67,00
29 Mo DATA 01,e4,2c,79,00,00,00,04,22,3c
30 re DATA 00,00,00,0c,24,3c,00,00,03,ed
31 FA DATA 4e,ae,ff,e2,23,e0,00,00,00,08
32 eJ DATA 67,00,01,e4,22,39,00,00,00,08
33 rV DATA 24,3a,00,00,00,28,61,00,02,2c
34 rF DATA 4e,ae,ff,d0,22,39,00,00,00,08
35 0F DATA 24,3c,00,00,00,0c,76,32,4e,ae
36 nY DATA ff,d6,53,00,67,00,01,98,41,f9
37 rC DATA 00,00,00,0c,42,30,00,00,22,39
38 HW DATA 00,00,00,08,24,3c,00,00,00,3d
39 0I DATA 61,00,01,f6,4e,ae,ff,d0,22,39
40 pN DATA 00,00,00,08,24,3c,00,00,00,48
41 zH DATA 76,32,4e,ae,ff,d6,53,00,67,00
42 XX DATA 01,62,41,f9,00,00,00,48,42,30
43 ex DATA 00,00,22,39,00,00,00,08,24,3c
44 rO DATA 00,00,00,54,61,00,01,e0,4e,ae
45 61 DATA ff,d0,22,39,00,00,00,08,24,3c

```

```

46 re DATA 00,00,00,46,76,00,4e,ae,ff,d6
47 2c DATA 53,00,66,04,7e,01,60,32,41,f9
48 7e DATA 00,00,00,86,41,f0,00,00,7e,00
49 rI DATA 7c,01,e4,2f,07,20,9e,3c,00,30
50 9F DATA 1c,20,67,18,9c,3c,00,30,cc,fc
51 rF DATA 0c,0a,de,4f,1e,20,67,0a,9c,3c
52 f4 DATA 00,30,cc,fc,00,64,de,4e,43,c7
53 rO DATA 00,00,00,6c,22,3c,00,00,00,0c
54 3e DATA 24,3c,00,00,03,ed,4e,ae,ff,e2
55 6c DATA 23,00,00,00,00,00,00,00,00,48
56 rH DATA 22,39,00,00,00,8c,24,3c,00,00
57 rV DATA 00,94,76,26,4e,ae,ff,d6,41,f9
58 7c DATA 00,00,00,00,00,00,00,00,00,20
59 7c DATA 66,00,00,b6,22,39,00,00,00,8c
60 67 DATA 34,39,00,00,00,86,53,42,e4,fc
61 j4 DATA 00,00,00,00,4e,ae,ff,e2,23,39
62 75 DATA 00,00,00,00,2c,3c,00,00,00,94
63 1Y DATA 26,3c,00,00,02,00,4e,ae,ff,d6
64 rA DATA 22,39,00,00,00,8c,4e,ae,ff,dc
65 0V DATA 00,00,00,00,00,22,3c,00,00,00
66 rY DATA 00,48,24,3c,00,00,03,ee,4e,ae
67 rA DATA ff,e2,23,e0,00,00,00,90,67,00
68 1 DATA 00,5e,41,f9,00,00,02,94,42,98
69 rF DATA 42,98,20,fc,00,00,00,04,7a,20
70 3q DATA 20,00,20,00,3c,00,00,1c,00,2c
71 rV DATA 00,00,00,00,22,39,00,00,00,30
72 rG DATA 24,3c,00,00,02,00,76,1e,ae,ae
73 bP DATA ff,00,22,39,00,00,00,00,00,2c,3c
74 BD DATA 00,00,00,94,26,3c,00,00,02,00
75 Wr DATA 4e,ae,ff,d0,22,39,00,00,00,00
76 rA DATA 4e,ae,ff,dc,42,b9,00,00,00,90
77 rI DATA 60,00,fe,40,4a,b9,00,00,00,90
78 rV DATA 00,00,00,79,00,00,00,00,22,39
79 rA DATA 00,00,00,00,00,00,00,00,00,b9
80 1V DATA 00,00,00,00,00,1c,2c,00,00,00
81 1v DATA 00,04,22,39,00,00,00,8c,4e,ae
82 2S DATA ff,dc,4e,b9,00,00,00,08,67,10
83 80 DATA 2c,79,00,00,00,00,00,00,00,00
84 r DATA 00,08,4e,ae,ff,dc,4a,b9,00,00
85 19 DATA 00,04,67,0e,2c,78,00,04,22,79
86 1Y DATA 00,00,00,00,4e,ae,fe,62,41,b9
87 rA DATA 00,00,00,00,67,0e,2c,78,00,04
88 7 DATA 2c,79,00,00,00,00,4e,ae,fe,66
89 2r DATA 70,00,4e,79,20,42,76,00,4a,18

```

```

90 F2 DATA 67,04,52,83,60,fb,4e,75,00,00
91 1Y DATA 00,00,03,ee,00,00,00,05,00,00
92 7w DATA 00,01,00,00,00,2a,00,00,00,46
93 7w DATA 00,00,00,66,00,00,00,9e,00,00
94 GL DATA 00,82,00,00,00,32,00,00,00,02
95 03 DATA 00,00,00,24,00,00,00,36,00,00
96 17 DATA 00,00,00,00,56,00,00,00,60
97 se DATA 00,00,00,74,00,00,00,7a,00,00
98 rA DATA 0f,8c,00,00,00,96,00,00,00,aa
99 2v DATA 00,30,00,00,00,00,00,e2,00,00
100 Hs DATA 00,cc,00,00,00,e0,00,00,00,e6
101 vH DATA 00,00,00,0a,00,00,01,2c,00,00
102 q3 DATA 01,32,00,00,01,42,00,00,01,4a
103 2t DATA 00,00,01,52,00,00,01,5a,00,00
104 sG DATA 01,6a,00,00,01,74,00,00,01,86
105 7U DATA 00,00,01,8c,00,00,01,9c,00,00
106 rF DATA 01,a6,00,00,01,ac,00,00,01,bc
107 yw DATA 00,00,01,e6,00,00,01,e6,00,00
108 0b DATA 02,ee,00,00,01,f8,00,00,01,fe
109 HW DATA 00,00,02,0e,00,00,02,18,00,00
110 5D DATA 02,22,00,00,02,2a,00,00,02,30
111 TA DATA 00,00,02,3a,00,00,02,42,00,00
112 1e DATA 02,48,00,00,02,52,00,00,02,5a
113 Bq DATA 00,00,02,60,00,00,02,6a,00,00
114 JM DATA 02,76,00,00,02,80,00,00,02,8c
115 tr DATA 00,00,00,00,00,00,03,f2,00,00
116 kn DATA 03,ea,00,00,00,19,e4,6f,73,2e
117 rJ DATA 6c,69,62,72,61,72,79,00,43,4f
118 Xp DATA 4e,3a,30,2f,30,2f,36,34,30,2f
119 rJ DATA 35,30,2f,20,41,4f,45,43,6f,6e
120 g6 DATA 76,65,72,74,20,00,0e,61,6d,63
121 rD DATA 20,64,65,72,20,41,4f,45,2d,44
122 rB DATA 61,74,65,69,3a,20,00,4e,61,6d
123 rL DATA 65,20,64,65,72,20,42,41,53,49
124 bz DATA 43,2d,44,61,74,65,69,3a,20,00
125 af DATA 4f,62,6a,65,6b,74,6e,75,6d,6d
126 rY DATA 65,72,3a,20,00,00,00,00,03,f2
127 ex DATA 00,00,03,eb,00,00,00,ac,00,00
128 Re DATA 03,f2

```

Listing 1. Der Basic-Lader für »AOE-Convert«

Programmname: AOE-Convert

Computer: A500, A1000, A2000 mit Kickstart 1.2 oder 1.3

Sprache: Assembler

Bemerkung: -

```

; AOEConvert
; 9.2.1989
; Michael Bertsch

ExecBase      = 4
FindTask      = 294
WaitPort      = 384
GetMsg        = 372
ReplyMsg      = 378
CloseLibrary  = -414
OpenLibrary   = -552
Open          = -30
Close         = 36
Read          = -42
Write         = -48
Seek          = 66
current       = 0
readOnly      = 1005
newFile       = 1006

```

```

Move.l ExecBase,A6
Sub.l A1,A1
Jsr FindTask(A6)
Move.l D0,A4
Tst.l $ac(A4)
Bne.s Start
Lea $5c(A4),A0
Jsr WaitPort(A6)
Lea $5c(A4),A0

```

Listing 2. Für Assembler-Interessierte: das Quellprogramm von »AOE-Convert«

Jsr GetMsg(A6)
Move.l D0,WMMsg

Start:

```

Lea DOSName,A1          ; DOS-Library öffnen
Moveq #0,D0
Jsr OpenLibrary(A6)
Move.l D0,DOSBase
Pec Ende
Move.l DOSBase,A6
Move.l # Fenster,D1      ; Fenster öffnen
Move.l #readOnly,D2
Jsr Open(A6)
Move.l D0,FensterHandle
Beq Ende

```

Schleife:

```

Move.l FensterHandle,D1  ; Frage nach dem 1. Dateinamen
Move.l #AOEFileFrage,D2
Bar Länge
Jsr Write(A6)
Move.l FensterHandle,D1  ; 1. Dateinamen holen
Move.l #AOEFileName,D2
Moveq #50,D3
Jsr Read(A6)
Subq.b #1,D0
Beq Ende
Lea AOEFileName,A0
Clr.b (A0,D0)
Move.l FensterHandle,D1  ; Frage nach dem 2. Dateinamen
Move.l #BASICFileFrage,D2
Bar Länge
Jsr Write(A6)
Move.l FensterHandle,D1  ; 2. Dateinamen holen
Move.l #BASICFileName,D2
Moveq #50,D3
Jsr Read(A6)

```



```

Subq.b #1,D0
Beq Ende
Lea BASICFileName,A0
Clr.b (A0,D0)
Move.l FensterHandle,D1 ; Frage nach der Objektnummer
Move.l #ObjektnummerFrage,D2
Bsr Länge
Jsr Write(A6)
Move.l FensterHandle,D1 ; Objektnummer holen
Move.l #Nummer,D2
Moveq #4,D3
Jsr Read(A6)
Subq.b #1,D0
Bne.s Rechnen
Moveq #1,D7
Bra.s Weiter

```

```

Rechnen:
Lea Nummer,A0 ; Objektnummer errechnen
Lea (A0,D0),A0
Moveq #0,D7
Moveq #0,D6
Move.b -(A0),D7
Beq.s Weiter
Sub.b #48,D7
Move.b -(A0),D6
Beq.s Weiter
Sub.b #48,D6
Mulu #10,D6
Add.w D6,D7
Move.b -(A0),D6
Beq.s Weiter
Sub.b #48,D6
Mulu #100,D6
Add.w D6,D7

```

```

Weiter:
Move.w D7,Nummer
Move.l #AOEFileName,D1 ; AOE-Datei öffnen
Move.l #readOnly,D2
Jsr Open(A6)
Move.l D0,AOEFileHandle
Beq Ende
Move.l AOEFileHandle,D1 ; Objekt einlesen
Move.l #Puffer,D2
Moveq #38,D3
Jsr Read(A6)
Lea Puffer,A0
Cmp.l #"AOE ",(A0)
Bne Ende
Move.l AOEFileHandle,D1
Move.w Nummer,D2
Subq.w #1,D2
Mulu #640,D2
Moveq #current,D3
Jsr Seek(A6)
Move.l AOEFileHandle,D1
Move.l #Puffer,D2
Move.l #512,D3
Jsr Read(A6)
Move.l AOEFileHandle,D1 ; AOE-Datei schließen
Jsr Close(A6)
Clr.l AOEFileHandle
Move.l #BASICFileName,D1 ; BASIC-Datei öffnen
Move.l #newFile,D2
Jsr Open(A6)
Move.l D0,BASICFileHandle
Beq Ende
Lea Puffer2,A0 ; BASIC-Datei erstellen
Clr.l (A0)+
Clr.l (A0)+
Move.l #4,(A0)+
Moveq #32,D5
Move.l D5,(A0)+
Move.l D5,(A0)+
Move.w #24,(A0)+
Move.w #15,(A0)+
Clr.w (A0)+
Move.l BASICFileHandle,D1
Move.l #Puffer2,D2
Moveq #26,D3
Jsr Write(A6)
Move.l BASICFileHandle,D1
Move.l #Puffer,D2

```

```

Move.l #512,D3
Jsr Write(A6)
Move.l BASICFileHandle,D1 ; BASIC-Datei schließen
Jsr Close(A6)
Clr.l BASICFileHandle
Bra Schleife

```

```

Ende:
Tst.l BASICFileHandle
Beq.s BASICFileZu
Move.l DOSBase,A6
Move.l BASICFileHandle,D1 ; BASIC-Datei schließen
Jsr Close(A6)

```

```

BASICFileZu:
Tst.l ACEFileHandle
Beq.s ACEFileZu
Move.l DOSBase,A6
Move.l ACEFileHandle,D1 ; AOE-Datei schließen
Jsr Close(A6)

```

```

ACEFileZu:
Tst.l FensterHandle
Beq.s KeinFenster
Move.l DOSBase,A6
Move.l FensterHandle,D1 ; Fenster schließen
Jsr Close(A6)

```

```

KeinFenster:
Tst.l .asse
Beq.s KeineDOSLibrary
Move.l ExecBase,A6
Move.l DosBase,A1 ; DOS-Library schließen
Jsr CloseLibrary(A6)

```

```

KeineDOSLibrary:
Tst.l WBMsg
Beq.s KeineWBMsg
Move.l ExecBase,A6
Move.l WBMsg,A1
Jsr ReplyMsg(A6)

```

```

KeineWBMsg:
Moveq #0,D0
Rts

```

```

Länge:
Move.l D2,A0 ; Länge des Textes ermitteln
Moveq #0,D3

```

```

L01:
Tst.b (A0)+
Beq.s L02
Addq.l #1,D3
Bra.s L01

```

```

L02:
Rts
Data

```

```

DOSName: Dc.b "dos.library",0
Fenster: Dc.b "CON:0/0/640/50/ AOEConvert ",0
AOEFileFrage: Dc.b "Name der AOE-Datei: ",0
BASICFileFrage: Dc.b "Name der BASIC-Datei: ",0
ObjektnummerFrage: Dc.b "Objektnummer: ",0

```

```

Bss
Align

```

```

WBMsg: Dc.l 0
DOSBase: Dc.l 0
FensterHandle: Dc.l 0
AOEFileName: Blk.w 30
BASICFileName: Blk.w 30

```

```

Dc.w 0

```

```

Nummer: Blk.w 3
AOEFileHandle: Dc.l 0
BASICFileHandle: Dc.l 0
Puffer: Blk.l 128
Puffer2: Blk.w 13

```

```

End (c) 1989 M&T

```

Listing 2. (Schluß)

Programmname: Farb-Lader
Computer: A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache: Amiga-Basic
Bemerkung: Zum Einbau in eigene Programme

Programmautor: Andreas Lietz

```

1 $AO ' Einlesen der AOE-Datei und Einstellung der Bildschirmfarben
2 BS OPEN "AOE-Datei" FOR IN=1 AS 1 'Hier Namen der AOE-Datei
  einsetzen
3 WY a$ = INPUT$(6, #1) 'Dateireader-Dateireader
4 SY FOR i= 0 TO 15 'Farbdaten lesen
5 W2 a$ = INPUT$(1, #1)
6 W3 b$ = INPUT$(1, #1)
7 J4 PALETTE 1, ASC(a$) AND 15 /16, ASC(b$) AND 240 /256, (ASC (b
  $ AND 15 /16
8 ZP0 NEXT i
9 W1 CLOSE 1
(C) 1989 M&T
    
```

Listing 3. Dieses Programm stellt die richtigen Farben für ein Objekt ein

Programmname: BOB-Move
Computer: A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache: Amiga-Basic
Bemerkung: Bitte richtige Dateinamen einsetzen!

Programmautor: Andreas Lietz

```

1 LNO SCREEN 1,320,256,4,1
2 F5 WINDOW 4, "AOE-Converter", 30,30 -(150,100),15,1
3 40 'Einlesen der AOE-Datei und Einstellung der Bildschirmfarben
4 DU OPEN "AOE-Datei" FOR INPUT AS 1 'Hier Namen der AOE-Datei
  einsetzen
    
```

```

5 Ya a$ = INPUT$(6, #1) 'Dateireader-Dateireader
6 W0 FOR i= 0 TO 15 'Farbdaten lesen
7 W2 a$ = INPUT$(1, #1)
8 W3 b$ = INPUT$(1, #1)
9 W6 PALETTE 1, ASC(a$) AND 15 /16, (ASC(b$) AND 240 /256, (ASC (b
  $ AND 15 /16
10 W00 NEXT i
11 W1 CLOSE 1
12 W8 'Bewegen eines Objektes auf dem Bildschirm
13 W7 ON COLLISION DOOR000 Aufprallen
14 W6 COLLISION ON
15 WX OPEN "BOB-Datei" FOR INPUT AS 1 'Hier Namen der konvertier
  ten Datei einsetzen
16 W3 OBJECT SHAPE 1, INPUT$(LOF(1, 1
17 W0 CLOSE 1
18 WJ WGET.X 1,10
19 WT OBJECT.Y 1,20
20 W1 OBJECT.VX 1,30
21 W8 OBJECT.VY 1,30
22 W8 OBJECT ON
23 W1 OBJ "" START
24 W0 WHILE 1
25 W0 SLEEP
26 W1 WEND
27 W8 Aufprallen
28 W02 x = WINDOW 1
29 W1 WINDY 4
30 W0 i = COLLISION
31 W2 IF i=0 THEN RETURN
32 W1 j = COLLISION
33 W6 IF j=2 OR i = -4 THEN
34 W04 OBJECT.VX 1, -OBJECT.VY 1
35 W02 ELSE
36 W04 OBJECT.VY 1, -OBJECT.VY 1
37 W02 END IF
38 W1 OBJECT.START
39 W0 WINDOW x
40 W00 RETURN
(C) 1989 M&T
    
```

Listing 4. Ein Beispielprogramm, das mit konvertierten Objekten arbeitet

Zauberei mit Bildern

Von Peter Sparlinek

Den Begriff Textverarbeitung kennt jeder. Doch genau wie man Text verarbeiten kann, ist dies auch bei Bildern möglich. Bildverarbeitung ist ein interessantes Thema. Tolle Effekte lassen sich damit erreichen. Doch wie geht das? »IFF_Filter« zeigt es Ihnen. Sehen Sie sich doch einmal die Beispielbilder an. Bild 1 ist das Original, das wir mit verschiedenen Filtern behandelt haben. Die Ergebnisse sehen Sie in Bild 2 bis 4.

Filter spielen in der Bildverarbeitung eine große Rolle. Aus »normalen«, farbigen Bildern können Bilder mit Grautönen

Ob in Stein gemeißelt oder nur aus Umrissen bestehend. »IFF_Filter« macht aus Ihren Farbbildern neue fantastische Grafiken. Probieren Sie es doch einfach einmal aus.

Peter Sparlinek

Den ersten Kontakt zu einem Commodore PET 2001 hatte Peter Sparlinek im Jahr 1980. Nachdem Basic zu langsam wurde, erfolgte der Umstieg auf 6502-Assembler. Die nächsten Schritte waren ein VC20 (1983) und ein C64 (1985). Zwei Jahre später wurde ein Amiga 500 angeschafft, auf dem der Autor vornehmlich in C programmiert. Den Anstoß zu IFF_Filter erhielt Peter Sparlinek in seinem Informatikstudium.



gemacht werden, die ganz anders aussehen. Es gibt die verschiedensten Typen wie den Tiefpaßfilter, den Gradientenfilter, etc. Jeder dieser Filter hat eine andere Aufgabe. Der Tiefpaßfilter filtert, wie der Name schon sagt, die hohen Frequenzen aus einem Bild. Unter hohen Frequenzen versteht man bei Bildern den häufigen Wechsel von Farben. Nebenbei entsteht der Effekt der Unschärfe. Je größer der Filter, desto stärker wird die Unschärfe. Mit dem Gradientenfilter kann man sehr gut die Kanten von den Objekten des Bildes hervorheben, während Flächen mit gleicher Intensität (gleicher Farbe) dabei verschwinden. Man erhält also im neuen Bild die Farbübergänge.

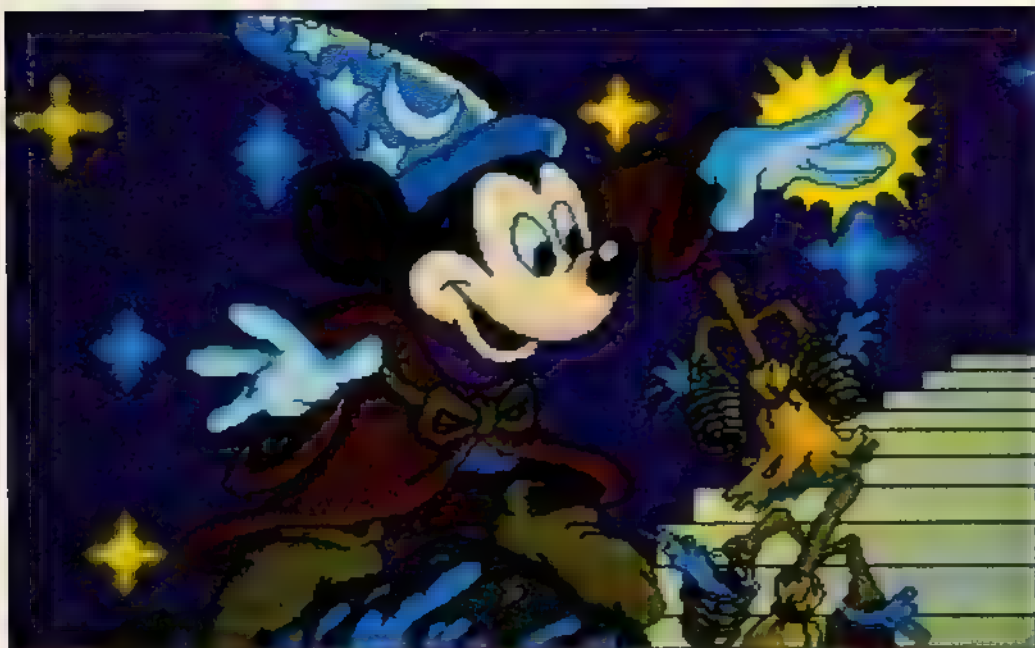


Bild 1. Das Originalbild mit einer Auflösung von 320 x 200 Punkten und 32 Farben

Filter – was ist das nun genau und wie funktioniert es? Diese Frage soll am Beispiel des 5x3-Tiefpaßfilters beantwortet werden. Der Tiefpaß berechnet den durchschnittlichen Grauwert in einer definierbaren (hier 5x3) Umgebung. Man kann ein Bild in Form einer Matrix darstellen, deren Elemente die Grauwerte beziehungsweise die Farbe enthalten. Ebenso kann man den 5x3-Tiefpaßfilter als Matrix darstellen.

Die Breite der Filtermatrix ist hier 5, die Höhe 3. Weil es ein Tiefpaßfilter sein soll, haben alle Elemente den Wert 1. Der zentrale Punkt der Filtermatrix ist der Mittelpunkt. Jetzt wird die Filtermatrix auf das Bild gelegt und alle Grauwerte des Bildes werden mit dem entsprechenden Wert der Filtermatrix multipliziert und die Produkte anschließend aufsummiert. Die Summe wird anschließend durch die Summe der Filtermatrixelemente (hier 15) dividiert und schon hat man den Grauwert, den das neue Bild an der Stelle unter dem zentralen Punkt annimmt. Dieses Verfahren wird für alle Bildpunkte angewandt.

Der Inhalt der Filtermatrixelemente legt fest, um was für einen Filter es sich handelt. Eine Übersicht über einige Filter sehen Sie in der Tabelle.

Wie man dem Namen nach schon vermuten kann, bearbeitet IFF_Filter Bilder, die im IFF-Standard vorliegen. Praktisch alle Malprogramme auf dem Amiga verwenden dieses Format. Dabei spielt es keine Rolle, ob ein Pinsel (Brush) oder ein Overscan-Bild (besonderes Grafikformat des Amiga) verarbeitet wird. Eine Ausnahme sind Bilder im HAM-Format, sie können nicht gefiltert werden.

IFF_Filter liefert als Resultat wieder ein IFF-Bild, man kann also die erzeugten Bilder mit den üblichen Programmen weiterbearbei-

ten. Während des Programmlaufs zeigt IFF_Filter die aktuelle Funktion an.

-n

Das erzeugte Bild wird negativ (nicht bei LOW und OWN).

-p

Das erzeugte Bild wird ungepackt gespeichert. Die Voreinstellung ist gepackte Speicherung. Sollte ein von IFF_Filter generiertes Bild nicht von Ihrem Malprogramm geladen werden, probieren Sie diesen Parameter.

-h<X>

Für <X> sind Werte von 1 bis 4 erlaubt. Das Eingabebild wird aufgehellt. Bei dem Wert 1 verwendet IFF_Filter normale Helligkeit. Setzen Sie 4 ein, wird das Bild stark aufgehellt.

-<typ>

Hiermit geschieht die Auswahl des Filters. Für <typ> können Sie folgende Parameter eingeben:

ID – Nur Umwandlung in Grautöne

REL1 – Relieffilter vertikal

REL2 – Relieffilter diagonal

GRA1 – Gradientenfilter diagonal

GRA2 – Gradientenfilter vertikal

GRA3 – Gradientenfilter alle 8 Richtungen

LOW – Tiefpaßfilter

OWN – selbst zu definierender Filter

Ein Beispiel verdeutlicht die möglichen Aufrufe:

IFF_Filter Seascape -h3 -p -REL1

Das IFF-Bild »Seascape« wird geladen und in ein Graubild umgewandelt. Dabei wird um den Faktor 3 aufgehellt. Nach dem Tief-

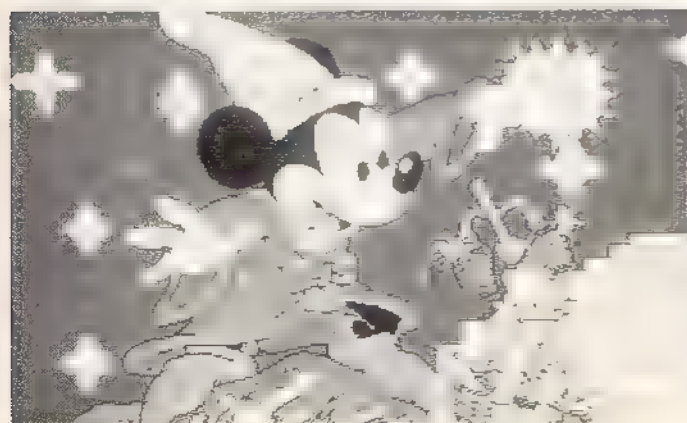


Bild 2. Das Originalbild (siehe Bild 1) nach der Umwandlung in Grautöne ohne Filter

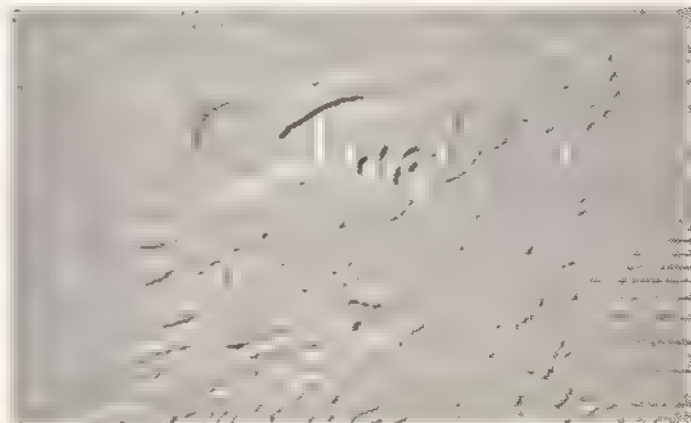


Bild 3. Das Originalbild (siehe Bild 1) bearbeitet mit dem Relieffilter 2 und Aufhellung um den Faktor 4

paß wird das gefilterte Bild ungepackt unter dem Namen »Seascape.Filter« gespeichert.

Natürlich sind auch Kombinationen der verschiedenen Filterarten verwendbar:

IFF_Filter DHO:Pic +i -dRAM:Bild -REL1 -GRA2

Diesmal lädt IFF_Filter das Bild »Pic« von der Festplatte DHO: und wendet den ersten Relieffilter und den zweiten Gradientenfilter darauf an. Dann speichert das Programm das Bild in der RAM-Disk unter dem Namen »Bild«. Während des Vorgangs werden Informationen zum Programmablauf ausgegeben.

Mit dem Parameter »OWN« sind Sie in der Lage, eigene Filter zu definieren. Dem Einfallsreichtum sind hier keine Grenzen gesetzt. Wählt man diese Option, fragt IFF_Filter zuerst nach der Größe der Filtermatrix. Diese muß ungerade sein, damit ein zentraler Punkt eindeutig festgelegt ist. Dies ist aber keine Einschränkung, denn man kann eine 2x2-Matrix in eine 3x3-Matrix einbetten, indem man die restlichen Zeilen und Spalten mit 0 füllt. Mit Ausnahme der Nullmatrix (alle Elemente sind 0) sind alle Matrizen erlaubt. In der Regel erzielen die Matrizen die besten Ergebnisse, die ausgewogen sind, das heißt die Summe aller Elemente ist Null. Für den eigenen Filter gilt, wie so oft, das Sprichwort: »Probieren geht über Studieren«.

René Beaupoil/kn



Bild 4. Das Originalbild (siehe Bild 1) bearbeitet mit dem Gradientenfilter 3 und Aufhellung (Faktor 4)

Filtermatrix		Bedeutung
1 1		Tiefpaß: Breite und Höhe der Matrix variabel (hier 7x3); Matrix mit 1 gefüllt. Unschärfe und Störpunkte verschwinden
+1 +4 +6 +4 +1 +2 +8 +12 +8 +2 0 0 0 0 0 -2 -8 -12 -8 -2 -1 -2 -6 -4 -1		Relieffilter 1 (vertikal): Bild wie aus Stein gemeißelt. Um den Relieffilter horizontal zu bekommen, muß die Matrix um 90 Grad gedreht werden
-1 -3 -3 -1 0 -3 -8 -6 0 +1 -3 -6 0 +6 +3 -1 0 +6 +8 +3 0 +1 +3 +3 +1		Relieffilter 2 (diagonal) wie bei Relieffilter 1, nur etwas anderer Blickwinkel
-1 0 0 +1		Gradientenfilter 1 diagonal: Hervorheben der Kanten in diagonalen Richtung
-1 +1		Gradientenfilter 2 vertikal: Hervorheben der Kanten in vertikaler Richtung
+1 +1 +1 +1 -8 +1 +1 +1 +1		Gradientenfilter 3 (alle 8 Richtungen): Hervorheben der Kanten in allen 8 Richtungen.

Tabelle. So sehen die verschiedenen Filtermatrizen aus, die in optimierter Form berechnet werden

Programmname: IFF_Filter

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: C

Compiler: Aztec-C V3.6

Aufrufe: CC IFF_Filter -E250 +L
LN IFF_Filter.o -LC32

Programmautor: Peter Sparlinek

```

1 ALO #include <stdio.h>
2 AJZ #include <exec/exec.h>
3 IO #include <libraries/dos.h>
4 LLG #define MIN(a,b) (a<b)?a:b
5 S2 #define ABS(x) (x>0)?(x):-(x)
6 CI #define INFO(s) if(info)puts(s)
7 7g long *MyLock;
8 Zr struct FileInfoBlock MyInfo;
9 2q long adr;
10 uo char *aktuell;
11 qb FILE *in,*out,*fopen();
12 11 int cc,i,j,k,filtertyp,fnsset=0,neg=0;
13 49 char *matrix,*erg,Zeile[100],string[60],text[5],c;
14 cU long help,colors[32];
15 G8 unsigned char bmd[25],jj,rot,blau,gruen;
16 LY int helligkeit,width,anzbyte,height,ddepth,depth,f_anz,line
s,info,count,pack,
planes,kompress=-10,
18 7MO char fo[]={0x46,0x4f,0x52,0x43,0x00,0x00,0x00,0x00,
19 HGE 0x49,0x4c,0x42,0x4d};
20 aX0 char cm[]={0x43,0x4d,0x41,0x50,0x00,0x00,0x00,0x30,
21 FAB 0x00,0x00,0x00,0x10,0x10,0x10,0x20,0x20,0x20,
22 OM 0x30,0x30,0x30,0x40,0x40,0x40,0x50,0x50,0x50,
23 1Y 0x60,0x60,0x60,0x70,0x70,0x70,0x80,0x80,0x80,
24 ce 0x90,0x90,0x90,0xa0,0xa0,0xa0,0xb0,0xb0,0xb0,
25 vF 0xc0,0xc0,0xc0,0xd0,0xd0,0xd0,0xe0,0xe0,0xe0,
26 NR 0xf0,0xf0,0xf0};
27 tLO /***** main *****/
*****/
28 *s main argc,argv
29 tT int argc;
30 V3 char **argv,
31 Ru {
32 QK1 int c;
33 Qc long zw;
34 11 char s[20];
35 FF helligkeit=1;filtertyp=0;text[4]=0;pack=1;info=0;
36 AN if(argc<2 || !strcmp(argv[1],"?") || 1>99) Hilfe();
37 o3 for(i=2;i<argc;i++)
38 Y1 {
39 Ha2 strcpy(s,argv[i]),
40 d8 if(!strcmp(s,"-D",2) || !strcmp(s,"-d",2))
41 t4 fnsset=1,
42 Gr3 strcpy(string,sa[2]);
43 oW
44 ID2 for(k=0;k<strlen(s);k++)s[k]=toupper(s[k]),
45 Vn if(!strcmp(s,"-H"))neg=1,
46 UZ else if(!strcmp(s,"+I"))info=1;
47 p8 else if(!strcmp(s,"-P"))pack=0;
48 1k else if(!strcmp(s,"-H",2))
49 xw
50 kD
51 CI3 if(strlen(s)==3)
52 92 if(s[2]>'0' && s[2]<'5') helligkeit=s[2]-'0';
53 rH2
54 sN1
55 x2 ReadIFF(argv[1]); /* Liest IFF-Bild ein */
56 j3 setres(erg,width*height,0x00);
57 rZ Col_to_grey(); /* Umwandlung in Grautöne */
58 90 for(i=2;i<argc;i++)
59 1W {
60 r2 strcpy(s,argv[i]);
61 13 for(k=0;k<strlen(s);k++)s[k]=toupper(s[k]);
62 3y if(!strcmp(s,"-ID") || id(),
63 a else if(!strcmp(s,"-REL1") || Relief1(),
64 EN else if(!strcmp(s,"-REL2") || Relief2 ),

```

Listing. Bildverarbeitung mit »IFF_Filter«. Bitte mit dem Checksummer (Seite 143) eingeben.


```

65 7j else if(!strcmp(s, "-GRA1"))Ableitung1();
66 Ou else if(!strcmp(s, "-GRA2"))Ableitung2();
67 P5 else if(!strcmp(s, "-GRA3"))Ableitung3();
68 PE else if(!strcmp(s, "-LOW"))Tiefpass();
69 nV else if(!strcmp(s, "-OWN"))Own();
70 8d1
71 tJ if(!fiset)
72 6Z ;
73 pa2 strcpy(string,argv[1]);
74 s0 strcat(string, ".Filter" );
75 A9 if(neg) strcat(string, "n"),
76 E31
77 8B if(!(out=fopen(string, "w"))) exit(puts("Kann Zieldatei nic
ht öffnen!\n"));
78 9n printf("Bild in Datei %s\n",string);
79 iX PutIFF( ,
80 On fclose(out);
81 7v FreeMem(matrix,width*height);
82 5G FreeMem(erg,width*height);
83 Lq0
84 U1 /***** Funktionen *****/
85 5b Hilfe() /* Hilftext bei falschen Aufrufparamete
rn */
86 8r
87 901 puts("USAGE IFF_Filter <filename> [Parameter... " );
88 vI puts "Parameter":,
89 JO puts("-d<filename> Festlegen der Ausgabedatei";
90 hA puts("-n negativ");
91 vD puts("-p IFF-Bild ungepackt abspeichern");
92 hQ puts("-i Info über Programmauf");
93 qP puts("-h<n> Aufheilen des Bildes 0<n<5");
94 3u puts(" 1: normal ... 4 sehr hell");
95 zR puts("-<typ> Auswahl des Filters");
96 JX puts("\tmit <typ>:");
97 LN puts("\t\tID\tIdentität");
98 4u puts("\t\tREL1\tRelief-Filter 1");
99 Pk puts("\t\tREL2\tRelief-Filter 2");
100 Ww puts("\t\tGRA1\tAbleitung diagonal");
101 P puts("\t\tGRA2\tAbleitung waag- und senkrecht");
102 3I puts("\t\tGRA3\tAbleitung in alle 8 Richtungen");
103 Lu puts("\t\tLOW\tTiefpass");
104 8U puts("\t\tOWN\teigener Filter");
105 pI exit(1000);
106 100
107 eU Id() /* Bild nur in Grauwerte umwandeln - kein Filter
*/
108 89
109 11 register int x;
110 8X INFO("Bild in Grauwerte - kein Filter");
111 8H if(neg)
112 8D
113 8X2 for(x=0;x<width*height;x++)
114 4K3 *(erg+x)=14- *(matrix+x);
115 8M1
116 r, else
117 P22 for(x=0;x<width*height;x++)
118 273 *(erg+x)= *(matrix+x);
119 9Q0
120 8M1 Relief1() /* ReliefFilter vertikal */
121 tM
122 8H1 long zwischen,dw=2*width;
123 42 int k,l,i1,i2,i3,i4,i5;
124 7b INFO("RELIEF-FILTER 1");
125 61 for(k=2;k<width-2,k++)
126 yR
127 e12 help=dw+k;
128 1T i1=matrix[x-2]+4*matrix[k-1]+6*matrix[k]+4*matrix[k+1]+2*
trix[k+2],
129 Q1 i2=matrix[help-width-2]+4*matrix[help-width-1]+6*matrix[he
lp-width]+4*matrix[help-width+1]+6*matrix[help-width+2];
130 VE i3=matrix[help-2]+4*matrix[help-1]+6*matrix[help]+4*matr
ix[help+1]+matrix[help+2];
131 4J i4=matrix[help+width-2]+4*matrix[help+width-1]+6*matrix[he
lp+width]+4*matrix[help+width+1]+6*matrix[help+width+2];
132 3o i5=matrix[help+dw-2]+4*matrix[help+dw-1]+6*matrix[help+dw
]+4*matrix[help+dw+1]+matrix[help+dw+2];
133 8s for(l=2;l<height-2,l++)
134 6Z
135 8B3 zwischen=-i1-i2-i3+i4+i5;
136 9g if(neg)

```

```

137 7r4 zwischen=7-(zwischen*7)/672;
138 13 else
139 14 zwischen=7*(zwischen*7)/672;
140 83 erg[k*width*1]=(char)zwischen;
141 7Y i1=i2,i2=i3,i3=i4,i4=i5;
142 7, help=help+dw+2;
143 6Z i5=matrix[help-2]+4*matrix[help-1]+6*matrix[help]+4*matr
ix[help+1]+matrix[help+2];
144 8p2
145 4q1
146 8r0 /* Relief 1 */
147 8B Relief2() /* ReliefFilter diagonal */
148 8B {
149 61 int k,l,m,dw=width*2;
150 5 INFO("RELIEF-FILTER 2 (diagonal)");
151 82 for(x=2,x<width-2,x++)
152 7 {
153 832 for(l=2;l<height-2,l++)
154 83
155 13 help=k+l*width;
156 13 m=matrix[help-dw-2]+3*matrix[help-dw-1]+3*matrix[help-d
w]+3*matrix[help-dw+1]
157 84 -3*matrix[help-width-2]-5*matrix[help-width-1]-6*matrix
[help-width]+matrix[help-width+2];
158 833 m+=3*matrix[help-2]-6*matrix[help-1]+6*matrix[help+1]+3
*matrix[help+2]
159 84 -6*matrix[help+width-2]+6*matrix[help+width]+8*matrix[hel
p+width+1]+3*matrix[help+width+2];
160 833 z=matrix[help-dw-1]+3*matrix[help-dw]+3*matrix[help+dw-
1]+matrix[help+dw+2];
161 83 if(neg)
162 84 m=7-(m*7)/490;
163 833 else
164 84 m=7+m*7/490;
165 833 erg[k*width*1]=(char)m;
166 833 }
167 833 }
168 833 }
169 833 }
170 833 }
171 833 }
172 833 }
173 833 }
174 833 }
175 833 }
176 833 }
177 833 }
178 833 }
179 833 }
180 833 }
181 833 }
182 833 }
183 833 }
184 833 }
185 833 }
186 833 }
187 833 }
188 833 }
189 833 }
190 833 }
191 833 }
192 833 }
193 833 }
194 833 }
195 833 }
196 833 }
197 833 }
198 833 }
199 833 }
200 833 }
201 833 }
202 833 }
203 833 }
204 833 }
205 833 }
206 833 }
207 833 }
208 833 }
209 833 }
210 833 }
211 833 }
212 833 }

```



```

212 Mp      ne p=z-1*width,
213 pX3      m=matrix[help*width-1-matrix[help*width+1]+matrix[help+
214 Lo        +matrix[help+width-1]+matrix[help-width]+matrix[help+1]
215 p34      +matrix[help+width]-matrix[x-seg-i]-8*matrix[help],
216 v7        if(neg)
217 b21      m=7-m*7/112,
218 p34      else
219 M41      m=7+m*7/112,
220 vt4      erg[k+width*1]=(char)m;
221 v3
222 b52
223 b61
224 v70
225 r2      Tierpass
226 a3
227 Jo      int i,j,s,x,y,x1,y1,xdiv,ydiv,ywidth,xy,hh,ok;
228 bf      char c;
229 vQ      puts("TIERPASS");
230 wh      for(ok=0,ok==0,
231 f8      ]
232 hq2      puts("\nBitte Größe des Filters eingeben. (Nur ungerade W
233 vV      erte)");
234 p8      printf("waagrecht: (3<= x <=99) "); scanf("%d",&x1);
235 p8      printf("senkrecht: (3<= y <=17) "); scanf("%d",&y1);
236 no      if(x1>2 && y1>2 && x1<100 && y1<18 && (x1%2) && (y1%2
237 cu1      )) ok=1;
238 Zz      if(!fset)
239 x2      sprintf(string,"%s_xd_yd",string,x1,y1);
240 Bv1      xdiv=x1/2,ydiv=y1/2
241 Bq      xy=x1*y1,ywidth=ydiv*width,
242 ab      for(x=ywidth,x<width-ywidth,x++
243 hq2      for(y=x-ywidth,y<=x+ywidth,y--width)
244 ATJ      erg[x]=(char)(erg[x]+matrix[y]);
245 l1      for(x=0,x<width-x-
246 qJ2      for(y=ywidth-width-x,y< height-ydiv*width+x,y--width)
247 B53      erg[y]=(char)(erg[y-width]-matrix[y-width-ywidth]+matrix
248 ZP1      "y-ywidth");
249 p9      movmem(erg,matrix,width*height);
250 p9      setmem(erg,width*height,0x00);
251 wv      for(y=ywidth;y<(height-ydiv)*width;y--width)
252 vR      for(hh=x=0,x<x1,x-- hh=matrix[x+y,
253 Jg2      for(x=xdiv,x<width-xdiv,x--
254 l3      ]
255 o3      erg[x+y]=(char)(hh/xy);
256 IT      hh=hh-matrix[x+y-xdiv-1]-matrix[x+y-xdiv,
257 Bdz
258 gdt ]
259 Af0      /* Eigener Filter */
260 b6
261 m2      int *fil,xm=0,ym=0,xo,yo,zv,il,jj,neg,pos,kk,ll,x1,il,12;
262 130      restart
263 Bu1      while((xm>0 && xm<width/4) && !(ym>0 && ym<height/4))
264 f      ]
265 b2      printf("Bitte Filtergröße eingeben\nwaagrecht (0<x<=d)
266 BM      ",width/4);
267 27      scanf("%d",&xm);
268 IT      printf("senkrecht (<=<=d ",height/4);
269 q1      scanf("%d",&ym);
270 Jo      if(!xm%2) printf("x auf %d angehoben\n",++xm);
271 Jn      if(!ym%2) printf("y auf %d angehoben\n",++ym);
272 Jd      if (fil= int * malloc(sizeof(int)*xm*y)
273 Lo
274 1v2      puts("Kann nicht genügend Speicher belegen\nVersuchen Sie
275 CI      kleinere Werte");
276 Lx1      goto restart;
277 f8      pos=neg+0;
278 Lx      puts("Eingabe der Filtermatrix");
279 P1      for(ii=1,ii<=n,ii++
280 Xv2      for(jj=0,jj<=xm,jj++
281 Tw      ]
282 mX3      printf("\nFilter-%d,%d=",ii-1,jj-1; scanf("%d",&zv),
283 F6      if(zv>0) pos+=zv; else neg-=zv;
284 xt      fil[ii*xm+jj]=zv,
285 b62
286 6f1      if(pos==neg && neg==0)

```

```

287 Z2      {
288 Q2      puts("Nullmatrix nicht erlaubt!");
289 QW      goto restart;
290 gB1      ]
291 +s      puts("Filter ok ? (j/yn)");
292 ag      [char c;
293 vT2      for(c=' ',c!='y' && c!='j' && c!='n';c=getchar());
294 b1      if(c=='n') goto restart;
295 L61
296 p8      INFO("Berechnung des gefilterten Bildes. Es kann etwas dau
297 v1      ern ....");
298 b6      pos=neg,
299 b6      for(il=xm/2,il<width-xm/2,il++
300 L12      [
301 L6      for(jj=width*ym/2+il,jj<width*(height-ym/2+il);jj+=width
302 L6      th)
303 L6      sw=0,il=xm-xm/2+width*ym/2+(width*ym/2)-xm/2,
304 L6      for(kk=il-1,kk<=kk<=il+width*ym/2+width
305 p4      for(ll=kk, ll<=kk<=ll+width*ym/2+width
306 p5      zw= *fil-xf *matrix[il,
307 p3      erg[jj]=(char)(zw+neg*14/pos,
308 L82
309 L1
310 L8
311 L8      free(fil,
312 En      Col_to_gray() /* Grauwertumwandlung */
313 p8
314 J1      register int il,jj,kk;
315 PX      INFO("GRAUWERTUMWANDLUNG");
316 L6      kk=width*height,
317 v6      for(il=0,il<=kk,il++
318 vY2      matrix[il]= char colors[matrix[il]],
319 ad0
320 vV      ReadIFF(bildname) /* Bild im IFF-Format einlesen */
321 v6      char *bildname;
322 v6
323 v6      int body_gelesen=0,loop,lw,laenge,breite,
324 ip      long lang;
325 p8      MyLock=Lock(bildname,ACCESS_READ);
326 p8      if(!MyLock) exit(printf("Illegal Filename %s\n",bildname));
327 dp      if(!Examine(MyLock,&MyInfo)) exit(printf("Can't get %s\n",
328 p8      bildname));
329 ny      laenge=MyInfo.file_size-1,
330 az      UnLock(MyLock);
331 L6      aktuell=adr=AllocMem(laenge,MEMF_CHIP);
332 vY      in=Open(bildname,MODE_OLDFILE);
333 GM      Read(in,adr,laenge);
334 n1      Close in,
335 a3      INFO("Lesen IFF-Bild");
336 ze      while body_geleser
337 L6
338 L2      MyRead(text,4);
339 p8      if(!strncmp(text,"BMHD",4))
340 L6
341 L6      MyRead(bmhd,24); /* BitMapHeader einlesen */
342 L6      breite=bmhd[4]*256+bmhd[5]
343 L6      if(breite%16)width=((breite>>4)+1)<<4;
344 L6      else width=breite;
345 p8      height=bmhd[6]*256+bmhd[7];
346 L6      depth=char bmhd[12];
347 L6      kompress=(char)bmhd[14];
348 b1      ddepth=1<<depth; /* ddepth =2*depth */
349 a6      if(!matrix= malloc(width*height*MEMF_CLEAR)
350 L6      exit(puts("Kann nicht genügend Speicher belegen!"));
351 v3      if(erg=AllocMem(width*height,MEMF_CLEAR)
352 L6      FreeMem(matrix,width*height,
353 L6      exit(puts("Kann nicht genügend Speicher belegen!"));
354 L6
355 L6      if(info)
356 L6      puts("Bilddaten: ",
357 L6      printf("-Breite :%d\n-Hoehe :%d\n-Bitplanes :%d
358 L6      \n-Farben :%d\n-gepackt :",breite,height,depth,dde
359 L6      pth);

```

Listing. Bildverarbeitung mit »IFF_Filter« (Fortsetzung)


```

358 0X      if(kompress) puts("Ja"); else puts("Nein");
359 113
360 02      } /* BMHD */
361 01      else if(!strcmp(text,"CMAP",4))
362 0P
363 273      INFO("Farbtabelle (CMAP)");
364 7      MyRead(&f_ang,4);
365 00      f_ang =
366 4r      for(loop=0;loop<f_ang;loop++)
367 8
368 4      MyRead(&v,1);
369 2      MyRead(&gruen,1);
370 11      MyRead(&blau,1);
371 2      cc=(char){(((rot>>4)+(gruen>>4)+(blau>>4))*
372 17      16+filtertyp*255);
373 4      colors[loop]=(char){MIN(cc,0x00+ifiltertyp)};
374 03
375 52      * CMAP *
376 00      else if(!strcmp(text,"BODY",4))
377 00
378 03      INFO("Bild Daten (BODY)");
379 00      if(kompress<0) exit(puts("Kein BMHD-Chunk!"));
380 3      MyRead(text,4); /* CHUNK-Länge überlesen */
381 01      if(!kompress)
382 00
383 124      INFO("einlesen\n");
384 01      for(lines=0;lines<height;lines++)
385 20
386 005      lw=lines*width;
387 00      for(planes=1;planes<ddepth;planes<=<=1)
388 00
389 66      MyRead(erg,width/8);
390 04      for(i=0;i<width;i++)
391 607      for(k=128;k>0;j++,k>=1)
392 96      if(erg[i]&k) matrix[lw+j]*=planes;
393 135
394 004
395 003      else
396 00
397 124      INFO("Entpacken");
398 04      for(lines=0;lines<height;lines++)
399 00
400 00
401 725      lw=lines*width;
402 00      for(planes=1;planes<ddepth;planes<=<=1)
403 00
404 06      i=0;
405 0A      while i<width
406 0
407 97      MyRead(&v,1);
408 0E      if(v>0&&i<128
409 00
410 08      MyRead(erg[i],j-1);
411 0
412 097      i=j-1
413 0
414 0
415 0
416 0
417 0
418 0
419 0
420 0
421 0
422 0
423 0
424 0
425 0
426 0
427 0
428 0
429 0
430 0
431 0
432 0
433 0
434 0
435 0
436 0
437 0
438 0
439 0
440 0
441 0
442 0
443 0
444 0
445 0
446 0
447 0
448 0
449 0
450 0
451 0
452 0
453 0
454 0
455 0
456 0
457 0
458 0
459 0
460 0
461 0
462 0
463 0
464 0
465 0
466 0
467 0
468 0
469 0
470 0
471 0
472 0
473 0
474 0
475 0
476 0
477 0
478 0
479 0
480 0
481 0
482 0
483 0
484 0
485 0
486 0
487 0
488 0
489 0
490 0
491 0
492 0
493 0
494 0
495 0
496 0
497 0
498 0
499 0
500 0
501 0
502 0
503 0
504 0
505 0
506 0
507 0
508 0
509 0
510 0
511 0
512 0
513 0
514 0
515 0
516 0
517 0
518 0
519 0
520 0
521 0
522 0
523 0
524 0
525 0
526 0
527 0
528 0
529 0
530 0
531 0
532 0
533 0
534 0
535 0
536 0
537 0
538 0
539 0
540 0
541 0
542 0
543 0
544 0
545 0
546 0
547 0
548 0
549 0
550 0
551 0
552 0
553 0
554 0
555 0
556 0
557 0
558 0
559 0
560 0
561 0
562 0
563 0
564 0
565 0
566 0
567 0
568 0
569 0
570 0
571 0
572 0
573 0
574 0
575 0
576 0
577 0
578 0
579 0
580 0
581 0
582 0
583 0
584 0
585 0
586 0
587 0
588 0
589 0
590 0
591 0
592 0
593 0
594 0
595 0
596 0
597 0
598 0
599 0
600 0
601 0
602 0
603 0
604 0
605 0
606 0
607 0
608 0
609 0
610 0
611 0
612 0
613 0
614 0
615 0
616 0
617 0
618 0
619 0
620 0
621 0
622 0
623 0
624 0
625 0
626 0
627 0
628 0
629 0
630 0
631 0
632 0
633 0
634 0
635 0
636 0
637 0
638 0
639 0
640 0
641 0
642 0
643 0
644 0
645 0
646 0
647 0
648 0
649 0
650 0
651 0
652 0
653 0
654 0
655 0
656 0
657 0
658 0
659 0
660 0
661 0
662 0
663 0
664 0
665 0
666 0
667 0
668 0
669 0
670 0
671 0
672 0
673 0
674 0
675 0
676 0
677 0
678 0
679 0
680 0
681 0
682 0
683 0
684 0
685 0
686 0
687 0
688 0
689 0
690 0
691 0
692 0
693 0
694 0
695 0
696 0
697 0
698 0
699 0
700 0
701 0
702 0
703 0
704 0
705 0
706 0
707 0
708 0
709 0
710 0
711 0
712 0
713 0
714 0
715 0
716 0
717 0
718 0
719 0
720 0
721 0
722 0
723 0
724 0
725 0
726 0
727 0
728 0
729 0
730 0
731 0
732 0
733 0
734 0
735 0
736 0
737 0
738 0
739 0
740 0
741 0
742 0
743 0
744 0
745 0
746 0
747 0
748 0
749 0
750 0
751 0
752 0
753 0
754 0
755 0
756 0
757 0
758 0
759 0
760 0
761 0
762 0
763 0
764 0
765 0
766 0
767 0
768 0
769 0
770 0
771 0
772 0
773 0
774 0
775 0
776 0
777 0
778 0
779 0
780 0
781 0
782 0
783 0
784 0
785 0
786 0
787 0
788 0
789 0
790 0
791 0
792 0
793 0
794 0
795 0
796 0
797 0
798 0
799 0
800 0
801 0
802 0
803 0
804 0
805 0
806 0
807 0
808 0
809 0
810 0
811 0
812 0
813 0
814 0
815 0
816 0
817 0
818 0
819 0
820 0
821 0
822 0
823 0
824 0
825 0
826 0
827 0
828 0
829 0
830 0
831 0
832 0
833 0
834 0
835 0
836 0
837 0
838 0
839 0
840 0
841 0
842 0
843 0
844 0
845 0
846 0
847 0
848 0
849 0
850 0
851 0
852 0
853 0
854 0
855 0
856 0
857 0
858 0
859 0
860 0
861 0
862 0
863 0
864 0
865 0
866 0
867 0
868 0
869 0
870 0
871 0
872 0
873 0
874 0
875 0
876 0
877 0
878 0
879 0
880 0
881 0
882 0
883 0
884 0
885 0
886 0
887 0
888 0
889 0
890 0
891 0
892 0
893 0
894 0
895 0
896 0
897 0
898 0
899 0
900 0
901 0
902 0
903 0
904 0
905 0
906 0
907 0
908 0
909 0
910 0
911 0
912 0
913 0
914 0
915 0
916 0
917 0
918 0
919 0
920 0
921 0
922 0
923 0
924 0
925 0
926 0
927 0
928 0
929 0
930 0
931 0
932 0
933 0
934 0
935 0
936 0
937 0
938 0
939 0
940 0
941 0
942 0
943 0
944 0
945 0
946 0
947 0
948 0
949 0
950 0
951 0
952 0
953 0
954 0
955 0
956 0
957 0
958 0
959 0
960 0
961 0
962 0
963 0
964 0
965 0
966 0
967 0
968 0
969 0
970 0
971 0
972 0
973 0
974 0
975 0
976 0
977 0
978 0
979 0
980 0
981 0
982 0
983 0
984 0
985 0
986 0
987 0
988 0
989 0
990 0
991 0
992 0
993 0
994 0
995 0
996 0
997 0
998 0
999 0

```

```

440 8x1      /* while *
441 00      FreeMem(sdr_senge);
442 20      ] /* ReadIFF */
443 00      PutIFF() /* Bild im IFF-Format abspeichern */
444 02
445 01      int lw;
446 0X      INFO("Schreiben des IFF-Bildes");
447 0V      for(lines=0;lines<height;lines++)
448 00
449 00      lw=lines*width;
450 0X      for(planes=1;planes<16;planes<=<=1)
451 0
452 003      cc=0;
453 01      for(i=0;i<=width;i++)
454 0
455 004      c<=<=1;
456 01      if(erg[lw+i]&planes) cc++;
457 07      if(!((i%8)&&1)
458 00
459 00      if(pack)
460 00      Zeile[i/8-1]=char;
461 05      else
462 006      matrix[count++]=(char)cc;
463 005      cc=0;
464 004
465 003      if(pack) Packen();
466 002
467 001      fo[6]=(char){(4+28+56+8+count+count%2)/256};
468 00      fo[7]=(char){(4+28+56+8+count+count%2)/256};
469 00      fwrite(fo,12,1,out); /* FORM-Chunk */
470 00      fwrite("BMHD",4,1,out);
471 00      bmd[14]=char;
472 00      bmd[12]=char;
473 00      fwrite(bmd,24,1,out); /* BMHD-Chunk */
474 00      fwrite(cm,56,1,out); /* CMAP-Chunk */
475 00      fwrite("BODY",4,1,out);
476 00      fwrite(0x00,out);fwrite(0x00,out);
477 00      fwrite((char){(count+count%2)/256},out);
478 00      fwrite((char){(count+count%2)/256},out);
479 00      fwrite(matrix,count,1,out);
480 00      if(count%2) fwrite(0x00,out); /* auf gerade Anzahl bringen
481 00
482 00      /* PutIFF *
483 00      Packen() /* IFF-Bild packen */
484 00
485 00      int last,i,j;
486 00      char m;
487 00      last=width/8;
488 00      i=0;
489 00      while i<last
490 00
491 00      m=Zeile[i];
492 00      for(j=i+1;j<=Zeile[j]&&j<last;j++)
493 00      if(j==i+1
494 00      { /* Jetzt wird gepackt */
495 00      matrix[count++]=(char){m+1};
496 00      matrix[count++]=(char){m};
497 00      i=j;
498 00
499 00      else
500 00      { /* Hier wird ungepackt übernommen */
501 00      for(m=Zeile[i];m<=last;m=Zeile[j++],
502 00      if(j<last j--2
503 00      j=MIN(last-1,j);
504 00      matrix[count++]=(char){j-1};
505 00      for(i=j,matrix[count++]=(char){Zeile[i++],
506 00
507 00      } /* while */
508 00      } /* Packen */
509 00      MyRead(nach,anz) /* Hilfsfunktion für ReadIFF() */
510 00      char *nach;
511 00      int anz;
512 00
513 00      int i;
514 00      for i=0;i<anz;i++ *nach++=*factive++;
515 00
516 00      C 1989 M&T

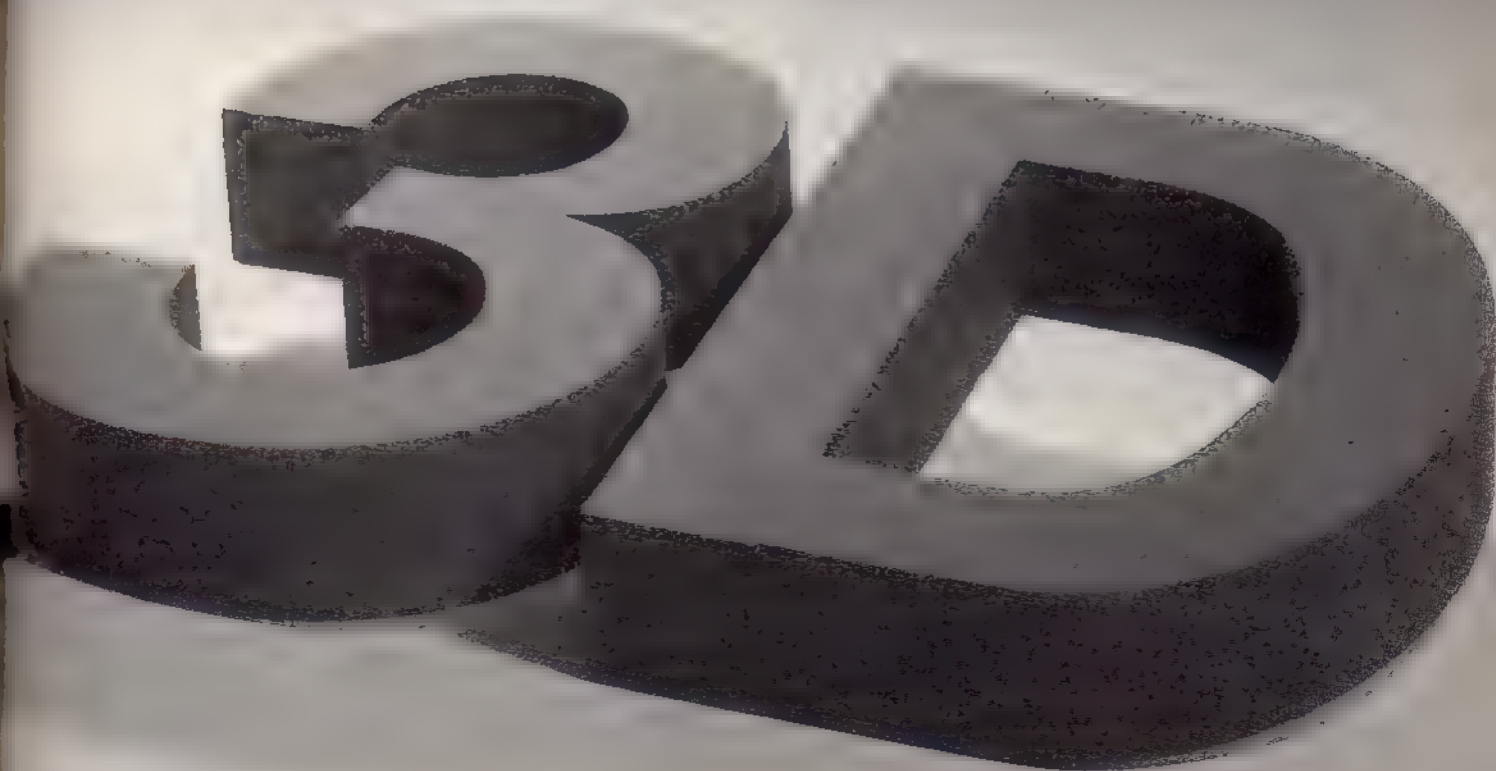
```

Listing. Bildverarbeitung mit «IFF_Filter» (Schluß)



Grafik **in der dritten** **Dimension**

»Ray Tracer« heißt wörtlich übersetzt »Strahlenverfolger«. Wer aber glaubt, mit dem folgenden Programm könne er seinen Amiga als Geigerzähler einsetzen, der irrt sich. Die Strahlen sind in unserem Fall harmloses Licht und verfolgt werden sie zur dreidimensionalen Darstellung.



AMIGA

SONDERHEFTE

1 Rund um den Amiga

Ausführliche Kurse,
Bauanleitung, Tips & Tricks

Das Sonderheft 1 enthält einen ausführlichen Basic-Kurs - alles über den richtigen Umgang der alternativen Benutzeroberfläche "Command Interface Line" - Bauanleitung. Mit dem Digitizer und der zugehörigen Software "Digisoft Plus" tauchen Sie ein in die Welt der digitalen Klänge. Tips & Tricks: aufgeteilt nach vier Bereichen: Einsteiger, Insider, Amiga-Basic und CU



2 Listings der Spitzenklasse

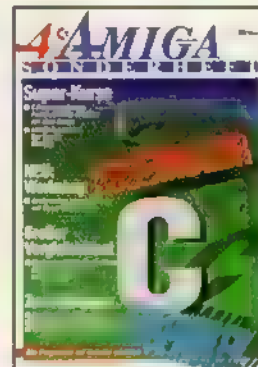
Tolle Anwendungen,
Tips, Tricks und Tools

"Fontdesigner", "Objekt Editor", "Haushaltsbuch", "Keyboard-Master", "Fast Load-Copy", "Super-Copy" sind nur einige von vielen Anwendungen im Sonderheft 2. Außerdem: Ratschläge zu Superbase und zwei außergewöhnliche Spiele

3 C und Assembler

Zwei Super-Kurse und
viele Tips und Tricks

Der große C-Kurs für alle Umstelter. Programmieren Sie ab sofort mit kräftiger Unterstützung durch Betriebssystem und Libraries. Der Assembler-Kurs enthält alle Geheimnisse des 68000er Prozessors



4 Basic und Spiele

10 Spitzenspiele zum
Abtippen in AMIGA-Basic

Die absoluten Spiele Top-Hits für den AMIGA in einer Übersicht. Mit Klassikern wie "Shanghai" und vielen brandneuen Hits. Ein ausführlicher Basic-Kurs führt zu Programmen mit rasanten Geschwindigkeiten

5 Public Domain

Software zum Nulltarif:
Das Beste aus aller Welt

Die ganze Faszination der Grafik in einem Projekt. Wichtigster Themenbereich werden Utilities mit zahlreichen Hilfsprogrammen



7 Spieleprogrammierung

Knowhow, Projekte, Spiele

Knowhow: Wir lüften die Geheimnisse von Blitter, Copper & Co. Projekte: Die erste 3D-Vektor Library für den Amiga ermöglicht völlig neue Erlebnisse in der Grafik-Welt. Lesen Sie über die Entstehung neuer Spiele von der Idee bis zum Programm. Spiele: Fünf brandheiße Spiele aus diesem Heft sind - neben vielen Tools und Utilities - auf Diskette erhältlich.



6 Anwendungen

Workshops: Experten verraten alle Tricks im Umgang mit den professionellen Programmen Sonix, Superbase und Beckertext. Überblick: Anwendungsprogramme und die Vorstellung der besten Programme. Top-Programme fast umsonst: Hochkarätige Anwendungen zur Textverarbeitung, Kontenführung und Dateiverwaltung bestechen durch ausgereifte Benutzerführung.

Ich bestelle: _____ Ausgaben AMIGA SH Nr. _____

_____ " " " " Nr. _____

_____ " " " " Nr. _____

_____ " " " " Nr. _____

_____ " " " " Nr. _____

Insgesamt: _____ Ausgaben für 16,-DM pro Exemplar

Name, Vorname _____

Straße, Hausnummer _____

PLZ, Wohnort _____

Rechnungssumme _____ DM

zzgl. 3,-DM Versandkosten _____ 3,-DM

Rechnungsbetrag insgesamt _____ DM

Schicken Sie bitte die ausgefüllte Karte zusammen mit dem Rechnungsbetrag als Verrechnungsscheck im Briefumschlag an Markt & Technik Verlag AG, Leserservice, Hans-Pinsel-Str 2 in 8013 Haar bei München

AMIGA
BESTELLCOUPON

Ray Tracing« nennt man die perspektivische Darstellung von dreidimensionalen Objekten unter Berücksichtigung der Lichtstrahlen. Was in diesem kurzen Satz an List und Tücke für den Programmierer steckt, wird erst vollends klar, wenn Sie sich verdeutlichen, was dabei berücksichtigt werden muß.

So müssen nicht nur die Abmessungen der Objekte (in unserem Fall Kugeln) der Perspektive entsprechend umgerechnet werden. Es muß auch der Lichteinfall berücksichtigt, die Spiegelungen richtig dargestellt (möglicherweise auf einer gekrümmten Oberfläche) und eventuell einzelne Flächen durchsichtig gemacht werden.

Verkürzte Rechenzeiten

Bei weniger leistungsfähigen Computern erweisen sich derartige Programme oft genug als wahre »Rechenzeitkiller«, die den Prozessor für erhebliche Zeit lahmlegen. Aber auch der Amiga – mit 68000er und dem Customchip Blitter sehr gut gerüstet – ist mit dieser Aufgabe stark gefordert.

So kann die Berechnung einer Grafik mit mehreren Elementen im Interlace-Modus durchaus zwei bis drei Stunden in Anspruch nehmen. Das heißt aber nicht, daß Sie sich einen zweiten Amiga kaufen müssen, weil Ihrer mit Raytracing demnächst ausgelastet ist. Es gibt einige Möglichkeiten, die Ausgabe zu beschleunigen oder notfalls abzubrechen.

Solange Sie an Ihrer Grafik noch Änderungen vornehmen, sollten Sie grundsätzlich im »Low-Resolution«-Modus arbeiten. Dieser ist etwa viermal so schnell wie der Interlace-Modus. Eine erhebliche Verkürzung der Rechenzeit bietet die Berechnung im »Test«-Modus. Diese stellt die Grafik im Briefmarkenformat dar. Sie brauchen auf eine Grafik nur wenige Minuten warten, für die sonst einige Stunden nötig wären.

Wollen Sie weitere Rechenzeit einsparen, ziehen Sie einfach während des Bildschirmaufbaus den Screen nach unten (oder stellen im Menü »verdeckt« ein). Sie müssen allerdings darauf achten, den Screen wieder ganz nach oben zu ziehen, bevor Sie mit »Window ändern« die Einstellungen bearbeiten. Denn dieser Menüpunkt öffnet ein eigenes Fenster, das sich nur durch ein Gadget am unteren Rand be-

seitigen läßt. Sollten Sie Ihren Amiga zwischendurch für dringende andere Aufgaben benötigen, so können Sie die Berechnung mit einem einfachen Mausklick an einer beliebigen Stelle des Screens abbrechen. Speichern Sie jetzt den derzeitigen Stand der Grafik. Wenn Sie weitermachen wollen, laden Sie diese und arbeiten an derselben Stelle weiter. Leider ist es uns aufgrund der Länge die-

Auflösung von 320 x 256 Punkten haben.

Berechnen: Hier finden Sie die beiden Punkte »Test« und »Start«. Bitte zügeln Sie vorerst Ihre Neugier und lassen Sie den Punkt »Start« links liegen. Dieser bringt längere Wartezeiten mit sich. Starten Sie statt dessen einen »Test«-Lauf. Dieser berechnet – weil noch keine Objekte definiert wurden – in wenigen Sekunden die Grafik.

```

  -*- Ray Tracer -*-

  Bildschirmmitte X = 0.000000
                  Y = 0.000000
                  Z = 17.999999
  Blickrichtung X = 5.000000
                  Y = 10.000000
                  Z = -2.600000
  Bildschirmabstand = 0.280000
  Brechungszahl = 2.560000
  »
  Kugelnummer [RET] = 0
  «
  Durchmesser = 50
  Kugelposition X = -50
                  Y = 200
                  Z = 200
  Sp/Gl/Li (1/2/3) = 3
  Kugel löschen (?) =
  OK
  
```

Bild 1. Mit dem Menüpunkt »W. ändern« stellen Sie alle wichtigen Daten für Ihre Grafik ein

ses Programms nicht möglich, das Listing abzdrukken, da sonst kaum Platz für weitere Programme in diesem Heft übrigbliebe.

Der Raytracer ist nur auf unserer Programmservice-Diskette enthalten, die Sie mit der Karte auf Seite 34 bestellen können.

Programmservice

Starten Sie den Raytracer entweder vom CLI mit »rt« und »RETURN« oder von der Workbench per Doppelklick auf das Icon. Sie sehen jetzt einen neuen Screen mit der Titelleiste. Drücken Sie die rechte Maustaste, dann sehen Sie vier Menüs: »RayTracer«, »Auflösung«, »Berechnen« und »Extras«.

Auflösung: Wählen Sie den Punkt »High« an, so wird der Bildschirm auf »High-Resolution« und Interlace umgestellt (640 x 512 Punkte).

Bevor Ihnen vom Flackern die Augen schmerzen, schalten Sie lieber zurück (mit dem Menüpunkt »Low«). Der Bildschirm sollte jetzt wieder die normale

Sie sehen jetzt in der linken oberen Ecke eine Grafik im Briefmarkenformat, die ein rot-grünes Schachbrettmuster und einen schmalen blauen Streifen als Horizont anzeigt.

Natürlich ist es nicht Sinn eines Raytracers, eine Landschaftsilhouette mit Schachbrettmuster auf den Bildschirm zu zaubern. Dafür gibt es weit anspruchslosere Programme.

Extras: Gehen wir also daran, das erste Objekt zu definieren. Zu diesem Zweck wählen Sie im Menü »Extras« den Punkt »W. ändern«. Sie sehen ein neues Fenster (Bild 1), in dem Sie alle wichtigen Parameter einstellen.

Sie sehen an der aktuellen Einstellung, daß ein Objekt bereits definiert ist. Es handelt sich dabei um die Kugel mit der Nummer Null, die als Lichtquelle dient. Sie können die Position dieser Lichtquelle ändern, indem Sie mit der Maus zweimal auf die entsprechenden Werte klicken und diese editieren, sobald der Cursor auftaucht.

Ähnlich gehen Sie vor, wenn Sie eine neue Kugel platzieren wollen. Klicken Sie zweimal auf die Nummer der Kugel (im Mo-

ment »0«) und tragen Sie dort eine Eins ein. Diese Änderung wird allerdings erst akzeptiert, wenn Sie mit einem Klick auf das Feld »[RET]« bestätigen. Sie sehen jetzt die aktuellen Maße Ihrer neuen Kugel angezeigt, nämlich fünf Nullen. Geben Sie ihr neue Werte, indem Sie die alten wie oben beschrieben ändern. Achten Sie darauf, daß Sie am Anfang Werte eintragen, die im darstellbaren Bereich liegen – später können Sie immer noch experimentieren.

Wir schlagen vor, daß Sie folgende Startwerte eintragen: Durchmesser 50, X-Position 50, Y-Position 100, Z-Position 200 und bei »Sp/Gl/Li« eine 2. Der letzte Wert unterscheidet zwischen spiegelnder (1) und glatter (2) Oberfläche beziehungsweise einer Lichtquelle (3).

Die erste eigene »Landschaft«

Klicken Sie mit der Maus jetzt auf »[RET]«, um die neuen Einstellungen zu bestätigen und danach auf das »ok«-Gadget am unteren Bildschirmrand.

Wenn Sie jetzt noch einmal »Test« auswählen, sehen Sie die Grafik mit Ihrer neu definierten Kugel am rechten Rand. Außerdem dürfte Ihnen aufgefallen sein, daß die Berechnung der Grafik jetzt deutlich mehr Zeit in Anspruch nimmt.

Um einen ersten Eindruck zu erhalten, sollten Sie Ihren Amiga vielleicht für zwei bis drei Stunden »beschäftigen«. Stellen Sie den Interlace-Modus ein (mit »Auflösung«/»High«), starten Sie die Berechnung (mit »Berechnen«/»Start«) und ziehen Sie den Screen ganz nach unten. Der Lohn des Wartens ist eine Grafik, die zwar leicht flackert (wegen Interlace), aber immerhin einen kleinen Eindruck von der Leistungsfähigkeit des Amiga vermittelt. Wenn Sie alles genau befolgt haben, sollte das Bild etwa so aussehen, wie das Foto im Inhaltsverzeichnis, das bei unseren ersten Versuchen entstand.

Wenn Sie nach diesem Vorgeschmack Lust zum Experimentieren bekommen haben, steht dem nichts mehr im Wege. Sehen Sie sich den Menüpunkt »W. ändern« etwas genauer an. Sie finden unter anderem Möglichkeiten, Ihren Blickwinkel (unter »Blickrichtung«), den Abstand vom Ort des Geschehens (mit »Bildschirmabstand«) und die Brechungszahl zu verändern. Experimentieren Sie mit diesen Werten. Wenn Sie eine Einstel-

lung gefunden haben, die Sie vollkommen zufriedenstellt, können Sie diese mit »E. sichern« speichern. Das Programm legt eine Datei »RTWerte« an, aus der beim nächsten Start die Voreinstellungen geladen werden.

RayTracer: Auf ähnliche Art können Sie fertige Grafiken speichern und wieder laden. Die entsprechenden Menüpunkte finden Sie unter »Raytracer« (»Laden« und »Saven«). Beide verzweigen in einen Filerequester, der nur Directories

oder selbstgespeicherte Grafiken erkennt. Dabei wird hinter jedem von Ihnen gewählten Namen entweder »-l« (wenn es sich um eine Low-Resolution-Grafik handelt) oder »-h« angehängt. Diese Erweiterung können Sie beim Speichern ebenso wie beim Laden weglassen.

Mit dem dritten Menüpunkt (»Ende«) sollten Sie vorsichtig umgehen. Er respektiert Ihren Wunsch und reagiert prompt, ohne vorher zu fragen, ob Sie nicht noch speichern wollen. Neben den bereits besproche-

nen Menüpunkten können Sie im »Extras«-Menü die Titelleiste ein- und ausblenden, ein Countdown-Fenster einrichten, das Sie informiert, wie viele Zeilen noch dargestellt werden müssen, und den aktuellen Screen sichtbar beziehungsweise unsichtbar machen. Auch diese Einstellungen werden in RTWerte gespeichert, so daß sie nicht jedesmal neu definiert werden müssen, sondern automatisch beim Start geladen werden. Das spart Zeit und erleichtert die Arbeit.

Mit den Informationen aus diesem Artikel und etwas Experimentierfreude sollten Sie in der Lage sein, fantastische Grafiken auf den Bildschirm zu zaubern. Wenn Sie mehr über die theoretischen Grundlagen dieses Themas wissen wollen, empfehlen das Buch »3-D-Grafik-Programmierung« von Data Becker, in dem Sie viele interessante Informationen zu diesem Gebiet finden. *Arno Götzler/kn*

3-D-Grafik-Programmierung Jennrich, Massmann, Schu. z. Data Becker, 284 Seiten ISBN 3-55011-174-2, Preis 59 Mark

Von Harry Kellermeyer und René Beaupoil

Mit »3DAnim« können Sie dreidimensionale Objekte mit Perspektive auf dem Bildschirm darstellen. Für diesen Zweck stellt Ihnen das Basic-Programm eine eigene Sprache zur Verfügung. 3DAnim arbeitet wie ein Interpreter und generiert Quellcode in Basic oder Assembler. Die Geschwindigkeit in Basic ist erstaunlich. Lassen Sie sich überraschen.

Um dreidimensionale Bewegungen zu beschreiben, braucht man ein Koordinatensystem mit drei Achsen und einem Ursprung. Dieser befindet sich direkt auf dem Monitor links unten. Nach rechts wird die x-Achse, nach oben die y- und gleichsam »in den Monitor hinein« die z-Achse angetragen. Ausgegangen wird von einem Quader, dessen vordere Seite die Monitorfläche ist und dessen Tiefe genauso groß ist wie die Breite. Die Einheit ist in alle Richtungen ein Hires-Punkt. Dadurch treten keine Verzerrungen auf (640 x 512 x 640 virtuelle Punkte), obwohl die eigentliche Animation am Ende bei 640 x 256 Punkten abläuft. Natürlich können auch Punkte außerhalb dieses Raums positioniert werden, nur sind diese nicht unbedingt sichtbar. Um die Zeit noch mit ins Spiel zu bringen, geht das Programm so vor: Die Bewegung wird, wie beim Daumenkino, in Einzelbilder aufgelöst. Diese Bilder werden dann nacheinander dargestellt, wobei sich nach dem letzten Bild das Ganze wiederholt.

3DAnim versteht 25 Befehle, die zur Erzeugung der Bewegung nötig sind. Sie werden im folgenden erläutert. Parameter in eckigen Klammern sind optional, müssen also nicht angegeben werden.

■ DIM p,n[opt]

Dieser Befehl muß der erste in der Liste sein und darf nur dort vorkommen. Es wird angegeben, wie viele Bilder man insgesamt benutzen will: DIM p,n bedeutet, daß auf n Bildern je p Punkte dargestellt werden sollen. Dabei geht die Numerierung von 0 bis p-1 und von 0 bis n-1. Wird der Parameter »opt« angegeben, werden keine Linien zum Fluchtpunkt gezeichnet (siehe auch PERS).

■ PERS x,y,z

Wenn Sie das Programm ausprobieren, stellen Sie fest, daß Ihr Computer außer Ihren Objekten noch zwei weitere Linien zeichnet. Es sind Orientierungslinien, die zum Fluchtpunkt führen, also zu dem Punkt, in dem alle zur z-Achse parallelen Linien scheinbar zusammenstoßen. Sie können den Fluchtpunkt durch diesen Befehl selbst bestimmen. Wenn Sie die beiden Li-

nien nicht im Bild haben wollen, geben Sie beim DIM-Befehl einfach einen dritten Parameter an (DIM 4,60,0).

■ ORIG x,y,z

Der Ursprung des Koordinatensystems wird neu definiert. Die Angaben beziehen sich immer auf das ursprüngliche Koordinatensystem, das heißt zweimal derselbe ORIG-Befehl bewirkt dasselbe wie nur einer. Es werden auch die Koordinaten aller Punkte auf den neuen Nullpunkt umgerechnet.

■ RES x[,y[,z]]

Mit diesem Befehl kann man im Koordinatensystem eine andere Auflösung wählen. Dabei können auch Verzerrungen entstehen. Fehlt ein Parameter, so wird er der Pixelgröße in x-Richtung angepaßt. Also entspricht »RES 320« dem Befehl »RES 320,256,320«.

Der Ursprung, Fluchtpunkt und sämtliche Punkte werden entsprechend umgerechnet, so daß sie auf dem Bildschirm an derselben Stelle erscheinen. Ist ein Parameter negativ, so kehrt sich die Richtung der Achse um.

■ INKS r1,g1,b1[,r0,g0,b0]

Die Parameter liegen im Bereich von 0 bis 15 und bestimmen die RGB-Werte der Farben 1 und 0. Wenn Sie die letzten drei Werte weglassen, werden sie automatisch 0, das heißt der Hintergrund ist schwarz. Voreinstellung ist »INKS 15,15,15,0,0,0«, also weiße Objekte auf schwarzem Hintergrund.

■ MASK flagx,flagy,flagz

Auf diese Weise kann man erreichen, daß bei den Befehlen SET, ACC, DEC, SHI, SHIL und COPY einzelne Komponenten ausmaskiert werden, wenn das entsprechende Flag 0 ist. Nach dem Kommando »MASK 1,1,0« werden durch einen COPY-Befehl nur noch die x- und y-Werte kopiert. Eine andere Anwendung ist, eine Rotation eines Körpers um eine zur x-Achse parallelen Achse mit entgegengesetzter Eigenrotation auf ein simples Auf und Ab zu reduzieren (durch Ausschalten der z-Komponente), um dies dann mittels »SHI dx,0,0« in einen Sinus zu verwandeln.

■ LET varwert

Einer Variablen (v0 bis v100) kann ein Wert zugewiesen werden.

■ SET p,x,y,z[,a[,b]]

Damit man Punkte bewegen kann, muß erst ein Anfang gemacht werden. Dies geschieht durch den Befehl SET. Dem Punkt p werden von Bild »a« bis Bild »b« die Koordinaten (x, y, z) zugewiesen. Fehlt b, so ist b gleich a. Ist weder a noch b vorhanden, gilt die Zuweisung für alle Bilder. Folgen mehrere SET-Befehle hintereinander und wird jeweils der nächste Punkt definiert, können die Parameter, durch einen Doppel-

Bewegung in Dimensionen

Bewegte, perspektivische, dreidimensionale Gittergrafiken möchten Sie? »3DAnim« liefert sie Ihnen. Je nach Geschmack in Basic- oder Assemblerquellcode.

Harry Kellermeyer

Der Autor begann mit seinem Computer-Hobby Ende 1985 auf einem Sharp PC 1401. Wegen der beschränkten Fähigkeiten folgte bald ein Schneider CPC 464. Im September 1987 schaffte sich Harry Kellermeyer dann einen Amiga 500 an. »3DAnim« war ursprünglich ein Programm, das einen rotierenden Oktaeder auf dem CPC 464 darstellte. Mit der größeren Leistungsfähigkeit des Amiga wuchs das Programm dann immer weiter. Das Resultat ist »3DAnim«, das vor Ihnen liegt. Mit dem Gewinn von 2000 Mark (Programm des Monats im AMIGA-Magazin 8/89) finanziert der Autor seinen Urlaub.

punkt getrennt, aneinandergehängt werden. Der jeweils erste Parameter muß dann aber vom zweiten Punkt an weggelassen werden. So wird aus

```
SET 4,1,1,1
SET 5,,8
SET 6,8,3,3,2
```

schließlich

```
SET 4,1,1,1,8:8,3,3,2
```

■ LINE punkteliste[:punkteliste[:punkteliste ...]]

Mit LINE wird festgelegt, in welcher Reihenfolge die Punkte jeweils miteinander verbunden werden sollen. Es werden zuerst die Punkte in der ersten Liste zusammenhängend gezeichnet, neu positioniert, weitergezeichnet und so weiter. Dabei besteht eine Punkteliste aus durch Kommata getrennten Parametern, die entweder ein Term oder ein Intervall (Term1..Term2) sind. So bedeuten beispielsweise:

```
5 Punkt 5
3,6 Punkte 3 bis 6
6,3 Punkte 6 bis 3 (in dieser Reihenfolge)
..10 Punkte 0 bis 10
4.. alle Punkte ab 4
.. alle Punkte
```

Kommen in einem Programm mehrere LINE-Anweisungen vor, so werden die einzelnen Punktelisten vor der Interpretation einfach aneinandergehängt und durch einen Doppelpunkt voneinander getrennt. Das heißt, die Befehle

```
LINE 2,1,3
LINE 4,5,6
```

bewirken dasselbe wie

```
LINE 2,1,3:4,5,6
```

■ REC Punkteliste:x0,y0,z0 ,dx,dy,dz

Den 4 Punkten in der Liste werden Koordinaten zugewiesen, so daß ein Rechteck entsteht. Der erste Punkt ist bei (x0, y0, z0), die restlichen ergeben sich, indem man gegen den Uhrzeigersinn weitergeht. Das Rechteck liegt immer parallel zu einer Koordinatenebene, es ist also dx, dy oder dz gleich Null.

■ LREC Punkteliste:x0,y0,z0 ,dx,dy,dz

Dieser Befehl verhält sich wie REC, nur werden zusätzlich die Punkte miteinander verbunden.

■ BOX Punkteliste:x0,y0,z0 ,dx,dy,dz

Mit diesem Befehl werden den 8 Punkten der Liste Koordinaten zugewiesen, so daß ein Quader entsteht. Die linke, untere Ecke liegt im Punkt mit den Koordinaten (x0, y0, z0). Der Quader hat die Maße dx, dy und dz.

■ LBOX Punkteliste:x0,y0,z0 ,dx,dy,dz

Dieser Befehl bewirkt genau dasselbe wie BOX, jedoch wird zusätzlich ein entsprechender LINE-Befehl ausgeführt.

■ MOVE Punkteliste:anfang,ende1[,ende2[,modus]]

Fehlen modus oder ende2, gilt modus=0, ende2=ende1. Innerhalb der Punkteliste kommt es, im Gegensatz zu LINE, nicht auf die Reihenfolge der Punkte an. Wiederholen sich Punkte, hat es auch keinen weiteren Einfluß. Voreinstellung ist »MOVE 0..p-1:0,n-1,n-1,0« (p,n=Anzahl aller Punkte und Bilder). Nach dieser Anweisung werden nur noch die Punkte in der Punkteliste bewegt. Danach folgende Teilanimationen beziehen sich nur noch auf das Bildintervall von »anfang« bis »ende2«, es wird die ganze Bewegung auf die Bilder von »anfang« bis »ende1« verteilt. Danach entscheidet »modus«, wie sie bis »ende2« verlängert wird.

modus=0: Die Bewegung wird genauso weitergeführt.

modus=1: Nach jeweils (ende1 + 1 - anfang) Bildern beginnt die Bewegung von vorne.

modus=2: Nach jeweils (ende1 + 1 - anfang) Bildern kehrt sich die Bewegung um.

Auswirkungen zeigt MOVE auf die Befehle ROT, ROTi, TURN, SHI, SHII, ACC, DEC, EXT.

■ SHI dx,dy,dz

Die Punkte werden linear ansteigend um insgesamt dx, dy und dz verschoben.

■ SHII dx,dy,dz

Hier werden die Punkte ebenfalls um dx, dy und dz verschoben, aber nicht allmählich, sondern in jedem Bild um diesen Betrag

■ ROT xm,ym,zm,xyw,yzw,dw

Dieser Befehl dient zum Drehen von Objekten und benötigt den Drehpunkt und drei Winkel. Sie müssen also erst einmal die Achse

im Raum konstruieren. Der Befehl

```
ROT 200,250,100,0,90,-360
```

bewirkt eine Drehung der Figur um den Punkt (200,250,100). Um die Drehachse zu erhalten, gehen Sie folgendermaßen vor:

Stellen Sie sich eine senkrechte Gerade durch den Drehpunkt vor. Diese wird nun um »xyw« gegen den Uhrzeigersinn parallel zur x-y-Ebene gedreht. Anschließend drehen wir diese Gerade noch um »yzw« Grad (wieder gegen den Uhrzeigersinn) parallel zur y-z-Ebene.

Die Drehung wird auf alle Bilder verteilt. Bei dem obenstehenden Befehl dreht 3DAnim bei insgesamt 120 Bildern das Objekt um jeweils 3 Grad pro Bild.

■ ROTi xm,ym,zm,xyw,yzw,dw

Für die Parameter gilt das gleiche wie beim ROT-Befehl. Der Unterschied ist der, daß in jedem Bild um den Winkel »dw« um die angegebene Achse gedreht wird und nicht wie bei ROT allmählich.

■ TURN xm,ym,zm,xyw,yzw

Auch hier gilt für die Parameter das gleiche wie beim ROT-Befehl, lediglich der Parameter »dw« fehlt. Was also die Bewegung angeht, so wird wie bei ROTi in jedem Bild um bestimmte Winkel gedreht, aber nicht um »dw« um die Achse, die konstruiert wird, sondern so, daß aus der ursprünglichen Achse (parallel zur y-Achse) die Konstruierte entsteht. Das heißt, es werden alle Punkte in jedem Bild um »wx« nach links und um »wz« zum Betrachter gekippt.

■ COPY quelle [,zielanfang] ,zielende

Die Punktkoordinaten des Bildes »quelle« werden auf die der Bilder »zielanfang« bis »zielende« kopiert. Wird der zweite Parameter weggelassen, so ist zielanfang gleich quelle+1 beziehungsweise quelle-1, wenn zielende kleiner als zielanfang ist.

■ COPY quellanfang,quellende,zielanfang,zielende

Die Koordinaten des Intervalls von quellanfang bis quellende werden auf die Bilder von zielanfang bis zielende übertragen.

■ ACC dx,dy,dz

Dieser Befehl entspricht ungefähr der SHI-Anweisung, jedoch wird nicht mit konstanter Geschwindigkeit verschoben, sondern in Ruhelage begonnen und dann konstant beschleunigt.

■ DEC dx,dy,dz

Analog zu ACC wird um dx, dy und dz verschoben, aber mit negativer Beschleunigung, das heißt die Punkte werden immer langsamer, bis sie schließlich ruhen.

■ EXT x0,y0,z0,faktor1,faktor2 [,anfang,ende]

Alle aktuellen Punkte erfahren dadurch eine Streckung mit dem Zentrum (x0,y0,z0). Der Abstand der Punkte dazu wird am Anfang jeweils mit faktor1 multipliziert, ab dann wächst der Faktor linear, bis er am Ende der Bewegung faktor2 ist. Sind die Faktoren -1, so hat man eine Punktspiegelung vor sich. Schaltet man durch den MASK-Befehl eine oder zwei Dimensionen aus, kann man die Streckung auf nur eine Richtung beschränken. Fehlen die letzten beiden Parameter, richtet sich die Bewegung nach dem letzten MOVE-Befehl (von anfang bis ende2).

■ FORM a,b

Die Punkte werden vom Ort im Bild »a« linear zum Ort im Bild »b« überführt. Man kann so zwei ähnliche Figuren definieren und dann ineinander übergehen lassen. Die alten Koordinaten zwischen den Bildern »a« und »b« gehen dabei verloren.

Die Befehle von 3DAnim kennen Sie nun, einige Fragen müssen jedoch noch geklärt werden. 3DAnim benötigt eine Befehlsliste im ASCII-Format, in der die einzelnen Anweisungen stehen, um die vollständige Animation zu berechnen. Die Liste der Befehle muß folgendermaßen aufgebaut sein: In jeder Zeile stehen ein oder mehrere Befehle, die durch einen senkrechten Strich (< SHIFT + \>) zu trennen sind, danach eventuell ein Strichpunkt mit nachfolgendem Kommentar. Jeder dieser Befehle kann mit beliebig vielen REM-Zeichen oder Leerzeichen beginnen. Einem Befehlswort muß ein Trennzeichen folgen (Ziffer, Rechen-, Leerzeichen, runde Klammer), ab da werden dann Leerzeichen und runde Klammern einfach ignoriert. Sie können so Koordinaten optisch zusammenfassen. Ob Sie Buchstaben groß oder klein schreiben, spielt keine Rolle. Wer den Basic-Editor verwenden will, kann durch ein REM-Zeichen (Alt + ä) am Zeilenanfang verhindern, daß der eingegebene Text verändert wird. Die Datei wird am Ende dann mit

```
SAVE "Name",A
```

gespeichert. Nach der Berechnung erzeugt das Programm mehrere Dateien auf Diskette: eine Datei, in der die berechneten Koordi-

naten für die Animation abgelegt sind (Namenserweiterung *.dat*) und Programme in verschiedenen Sprachen, die die Bewegung darstellen. Am Anfang werden Sie gefragt, in welchen Sprachen der Computer diese Programme erzeugen soll. In der abgedruckten Version stehen Ihnen Assembler und Basic zur Verfügung. Geben Sie nun bei der Eingabe eine Zeichenkette ein, die die Zeichen für die gewünschten Sprachen enthält. Sie sind auf dem Bildschirm aufgelistet. Kleinbuchstaben und entsprechende große bewirken dasselbe. Wenn Sie also »a« eingeben, wird Assemblercode erzeugt (Namenserweiterung *.asm*), der danach assembliert werden kann. Bei Eingabe von »b« schreibt Ihr Amiga ein Basic-Programm (Namenserweiterung *.bas*) auf Diskette. Wenn Sie »ab« eingeben, so befinden sich danach beide Programme auf Diskette.

Grundsätzlich versteht 3DAnim als Parameter einen gesamten mathematischen Term (zum Beispiel: $-9 \cdot 23.2 + x8/2 - v12 \cdot 10$). Zur Verfügung stehen Real-Zahlen (wie 1,4,12,-184,+-.5), Rechenzeichen (+,-,*,/,^), wobei »Punkt vor Strich« gerechnet wird. Außerdem gibt es einige Standardvariablen: »p« und »n« geben die Anzahl der Punkte beziehungsweise der Bilder an. »x«, »y« oder »z« mit nachfolgender Zahl (x0, y4 oder z38) gibt den Wert der entsprechenden Koordinate des Punktes mit dieser Nummer zurück. Dabei wird der Wert aus dem Bild genommen, mit dem die Bewegung beginnt.

Hat ein Parameter den Wert 0, kann man ihn auch weglassen: Aus »SET 1,0,2,0« wird »SET 1,,2«. Man sollte aber bei solchen Befehlen aufpassen, bei denen die Anzahl der Parameter eine Rolle spielt. Ist ein Parameter kleiner 0 oder größer n-1, so wird der Wert mittels »MOD« auf einen Wert dazwischen umgerechnet.

Sie können 3DAnim um andere Sprachen bei der Programmgenerierung erweitern. Wenn Sie etwa C und Pascal hinzufügen wollen, müssen Sie an den DATA-Befehl in Zeile 2 noch »c,C,p,Pascal« anhängen, des weiteren folgen ab Zeile 3 nun weitere Abfragen, hier also:

```
ELSEIF a$="C" THEN
  CALL C.code
ELSEIF a$="P" THEN
  CALL Pascal.code
```

Beim Vergleich von a\$ mit einem Zeichen muß dieses als Großbuchstabe dastehen. Nicht zuletzt ist natürlich der Programmteil nötig, der das Programm in der neuen Sprache auf Diskette schreibt, hier »C.code« und »Pascal.code«. So ein Programmteil wird einfach als Unterprogramm an 3DAnim angefügt. Dabei gibt es einiges zu beachten: Vor dem ersten READ muß der Zeiger durch RESTORE auf den Anfang des Unterprogramms gesetzt werden. Wie das Unterprogramm aufgebaut sein kann, können Sie ja am Beispiel von »Basic.code« sehen und analog programmieren. Wenn Sie keinen Assembler besitzen oder auf Assembler-Code (oder auf Basic-Code) verzichten können, lassen Sie einfach das ganze Unterprogramm weg und ändern noch die Zeilen 2 und 3 um.

Zum Assembler-Code: Ein eventuell erzeugtes Assembler-Programm ist so formuliert, daß es mit dem A68K-Assembler auf der Public Domain-Diskette Fish 110 übersetzt werden kann. Dazu kopieren Sie einfach das Programm (=Name.asm*) in die RAM-Disk und legen die Fish-Disk ein, danach compilieren Sie es vom CLI aus mit dem Befehl

```
a68k/a68k RAM:Name.asm
```

und linken die so entstandene Datei »RAM:Name.o« mit dem Kommando

```
pdc/examples/blink RAM:Name.o to RAM:Name.bin
```

zum lauffähigen Programm »RAM:Name.bin«. Dieses können Sie nun starten, wozu die Datei »Name.dat« auf der eingelegten Diskette stehen muß. Falls Sie diesen Assembler nicht besitzen, dürfte es kein Problem darstellen, 3DAnim entsprechend anzupassen. Falls doch Probleme auftreten, sollten Sie unbedingt die Leerzeichen in den DATA-Zeilen beachten.

Damit das generierte Basic-Programm läuft, müssen im selben Verzeichnis der Amiga-Basic-Interpreter, das Basic-Programm, die Datei mit den Daten und die zwei Dateien »graphics.bmap« und »Intuition.bmap« stehen. Wenn Sie einen Amiga mit 512 KByte Speicher besitzen, kann es vorkommen, daß bei den CLEAR-Anweisungen Probleme auftreten. Verkleinern Sie dann die angegebene Größe, bis das Programm läuft. Funktioniert dies nicht, müssen Sie leider die Anzahl der Bilder oder Punkte verringern.

Besitzer eines AC-Basic-Compilers können sich freuen. Sowohl 3DAnim als auch die generierten Programme lassen sich übersetzen. Achten Sie dabei jedoch darauf, daß die Schalter »R« und »N« gesetzt sind.

Zum Testen können Sie Listing 2 verwenden, das ein paar Fähigkeiten von 3DAnim zeigt.

Natürlich soll dieses Programm nicht professionelle ersetzen. Dennoch haben Sie damit eine relativ schnelle und preiswerte Möglichkeit, in die Welt der Animation hineinzuschnuppern. kn

Programmname:	3DAnim
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Bemerkung:	siehe Text
Programmautor: Harry Kellermeier	

1	130 -----
2	ON '----- 3D-ANIM -----'
3	zR '----- AMIGA -----'
4	'- Harry Kellermeier -'
5	'----- X -----'
6	'----- Init -----'
7	'A CLEAR,1500000
8	Ku ON BREAK GOSUB unterbr:BREAK ON
9	pk DEFINT i,j
10	WG DEF FNst(p1,p2)=(p2<p1)*2+1
11	HH DEF FNmd(p1)=(p1 MOD (n+1)+n+1)MOD (n+1)
12	! DIM ixx 5 ,linie 500 ,prere\$,r,n,xf,yf,flag3,stapelS
13	nK DIM v(100),p(8),zeichen0\$(15),sprache0\$(15)
14	yS p1=4*ATN x grad=p1,180
15	00 stapelS=CHRS 0
16	fg flag1=-1,flag=-0
17	i fx=1 fy=1 fz=1
18	u xx=-1 yy=-1 zz=-1
19	! crgx=c.orgy=c.orgz=0
20	VB xf=400 yf=300 zf=1600
21	rp lirieS="" sprache0\$=0:zeichen0\$=""
22	7P ixx 0=0,ixx 1=0 ixx 2=0
23	! ixx 3=15 ixx 4=15 ixx 5=15
24	!H '----- verfügbare Sprachen einlesen
25	kJ 2 DATA a,Assembler,b,Basic
26	kk DATA *,*
27	eh READ a\$,b\$
28	BY WHILE a\$<>"*"
29	zo2 sprache0\$=sprache0\$+1
30	hW zeichen0\$(sprache0\$)=a\$
31	!R zeichen0\$=zeichen0\$-CASE\$(a\$
32	0Q sprache0\$(sprache0\$)=b\$
33	!N READ a\$,b\$
34	zm0 WEND
35	30 '----- Screen aufbauen u. Programmname u. gewünschte Sprach
36	en einlesen
37	y2 SCREEN 1,640,256,2,2:WINDOW 2,,,0,1
38	qS PALETTE 0,0,0,1:PALETTE 1,0,0,1
39	Fr PALETTE 2,0,0,0:PALETTE 3,1,1,1
40	!U COLOR 3,2:CLS
41	9y FOR i=0 TO sprache0\$-1
42	!2 LOCATE 5+(1 MOD 5)*2,10+(1\5)*20
43	x3 PRINT zeichen0\$(i+1) "sprache0\$(i+1)
44	!N NEXT
45	En LOCATE 25,17:INPUT "Name der Quelldatei.... " p
46	name\$
47	01 WHILE zeichen1\$=""
48	bs2 LOCATE 27,17:PRINT SPACE\$(80)
49	!1 LOCATE 27,17:INPUT "Sprache(n) der Zieldatei(en).(s.o.).."
50	zeichen1\$
51	6H zeichen1\$=UCASE\$(zeichen1\$)
52	04 IF zeichen1\$<>" " THEN
53	Ko4 a\$=""
54	!A FOR i=1 TO LEN(zeichen1\$)
55	!G IF INSTR(zeichen0\$,MID\$(zeichen1\$,i,1)) THEN a\$=a\$+MID
56	!4 \$(zeichen1\$,i,1)
57	!I NEXT
58	!q zeichen1\$=a\$
59	!2 END IF
60	!9G WEND
61	!I CLS:LOCATE 1,26:PRINT "PROGRAMM: "pname\$.PRINT
62	!a POKER &HFF0*256-&B96,32
63	!e POKER &HFF0*256-&B14,0

Listing 1. Dreidimensionale Animationen mit »3DAnim«

AMIGA
SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt abtippen!

Die aktuellen Disketten zum Sonderheft 9:

Grafik

»Amiga-Objekt-Editor«: Überdimensionale Sprites spielend leicht entwickelt (genaue Programmbeschreibung auf Seite 6 in diesem Heft).
»AOE-Convert«: bauen Sie die mit dem Amiga-Objekt-Editor entwickelten Super-Sprites in Ihre Basic-Programme ein (Beschreibung S.18).
»IFF-Filter«: Farbbilder auf faszinierende Weise modifizieren (Beschreibung S.21).
»Ray Tracer«: Arbeiten mit dem Spiel von Licht und Schatten (Beschreibung S.21).
»3DAnim«: Gittergrafiken – bewegt, perspektivisch, dreidimensional (Beschreibung S.30).

Spiele

Flugsimulator (Beschreibung S.40)
Ping-Pong (Beschreibung S.50)
Krieg der Sterne (Beschreibung S.55)
Denkspiel Trainingslager (Beschreibung S.61)

Strategiespiel »Buh Au CS« (Beschreibung S.67)
Börsenspiel (Beschreibung S.76)

Tools

»MasterCruncher«: Packt alles (Beschreibung S.84).
»Virus Killer«: Unentbehrlich (Beschreibung S.90).
»TestDev«: Welche Devices sind vorhanden? (Beschreibung S.100).
Weiches Ein- und Ausblenden von Bildern (Beschreibung S.101).
»Shuffle«: Der schnelle Weg zum hintersten Fenster (Beschreibung S.102).
»Felder«: Beschleunigtes Abarbeiten von numerischen Feldern (Beschreibung S.104).

Anwendungen

»Keyboard-Master«: Lernprogramm für das 10-Finger-System (Beschreibung S.121).

»Sound-Effekt-Editor«: Geräusche selbst erzeugen (Beschreibung S.124).
Etiketten: Das Programm für Ihre Diskettenetiketten (Beschreibung S.130).
»hscroll«: Laufschriften erzeugen (Beschreibung S.136).
»Uni-Datei«: Universelle Dateiverwaltung mit komfortabler Benutzerführung (Beschreibung S.140).

Außerdem finden Sie alle Programme aus den Artikeln, die mit dem D-Symbol gekennzeichnet sind, auf den Programmservice-Disketten.

Bestell-Nr. 45009
Zwei Disketten!

DM 29,90*

sFr 24,90*/öS 299,-*

* Unverbindliche Preisempfehlung



**Weitere Angebote
auf der Rückseite!**

AMIGA PROGRAMMSERVICE

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie bei uns das Beste aus zwei Jahren Amiga-Magazin! Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

- »AmigaDat« – Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung
- »Manager« – Das komfortable Haushaltsbuch.
- »Disketik« – Drucken von Diskettenlabels
- »MasterCruncher« – Leistungsfähige Daten- und Programmkomprimierung.
- »Recover« – Retten von gelöschten Daten.
- »Resist« – Macht Programme resistent.
- »MouseCreator und PointerMaker« – Generieren Sie Ihre eigenen Mauszeiger.
- »DiskSpy« – Problemloses Ändern von Daten direkt auf Diskette.
- »AmigaSort« – Bringt Ordnung in Ihre Diskettensammlung
- »Fade« – Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.
- »VirusControl« – Der komfortable Virenschutz
- »TrapHandle« – Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action

- »Trooff« – Das spannende Motorradrennen der Zukunft.
- »Quadron« – Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen

- »Poker« – Wann bekommen Sie den Royal-Flush.
- »Billard« – Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.
- »Domino« – Verblüffend Umsetzung des bekannten Spiels.
- »Kniffel« – Eine grafisch verblüffende Würfelspielumsetzung für bis zu vier Spieler.
- »3D-Tic-Tac-Toe« – Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest

- »Eliza« – Der Amiga als Psychotherapeut
- »Arriba« – Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readme-File auf den jeweiligen Disketten.

Bestell-Nr. 47901
Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:

Bestellungen bitte nur gegen Vorauskassa an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Verlag AG, Kollerstrasse 37, CH 6300 Zug, Telefon (042) 440 550. Österreich: Markt&Technik Verlag Gesellschaft m.b.H. Große Neugasse 28, A-1040 Wien, Telefon (0222) 587 1393-0. Microcomputique, E. Schiller, Gogelstraße 17, A-3500 Krems, Telefon (02732) 741 93. MES-Verband, Postfach 15, A-3485 Haindorf, Büchereizentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 83 31 96. Bestellungen aus anderen Ländern bitte nur schriftlich an Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar. Nur gegen Bezahlung der Rechnung in voraus.

Bitte kein Bargeld einschicken!

Bitte verwenden Sie für Ihre Bestellung und Überweisung die beigeheftete Post giro-Zahlkarte, oder senden Sie uns einen Verrechnungsscheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Bit für Bit nur Hits...

...das Beste aus zwei Jahren

Amiga-Magazin!

Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

- »AmigaDat« – Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung
- »Manager« – Das komfortable Haushaltsbuch.
- »Disketik« – Drucken von Diskettenlabels
- »MasterCruncher« – Leistungsfähige Daten- und Programmkomprimierung.
- »Recover« – Retten von gelöschten Daten.
- »Resist« – Macht Programme resistent.
- »MouseCreator und PointerMaker« – Generieren Sie Ihre eigenen Mauszeiger.
- »DiskSpy« – Problemloses Ändern von Daten direkt auf Diskette.
- »AmigaSort« – Bringt Ordnung in Ihre Diskettensammlung
- »Fade« – Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.
- »VirusControl« – Der komfortable Virenschutz
- »TrapHandle« – Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action

- »Trooff« – Das spannende Motorradrennen der Zukunft.
- »Quadron« – Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen

- »Poker« – Wann bekommen Sie den Royal-Flush.
- »Billard« – Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.
- »Domino« – Verblüffend Umsetzung des bekannten Spiels.
- »Kniffel« – Eine grafisch verblüffende Würfelspielumsetzung für bis zu vier Spieler.
- »3D-Tic-Tac-Toe« – Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest

- »Eliza« – Der Amiga als Psychotherapeut
- »Arriba« – Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readme-File auf den jeweiligen Disketten.

Bestell-Nr. 47901

Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:

DM 39,90* sFr 39,-*/ öS 390,-*

Direkt bestellen statt abtippen!

Die aktuelle Diskette zum Heft:

Amiga-Sonderheft 7

Reflex: Das neueste Projekt von Arno Gölzer. »Reflex« ist der Nachfolger der beliebten Billardumsetzung. Lassen Sie sich von einer einzigartigen Spielidee und der tollen grafischen Umsetzung begeistern. Beschreibung auf Seite 66. Noch mehr starke Spiele... Ob Action- oder Strategiespiel – hier finden Sie alles, was das Spielerherz begehrt. »Fast Freddie« Die Beschreibung dieses Actionspiels finden Sie auf Seite 86. »Memory« Eine interessante grafische Umsetzung für Leute mit gutem Gedächtnis. Beschreibung auf Seite 97. »Reactor« Das Strategiespiel, das auch Atomkraftgegner fesseln wird. Beschreibung auf Seite 107. »Schiebespiel« Strategisches Denken und Tüfteln ist bei diesem Spiel das A und O. Beschreibung auf Seite 88.

Vector-Grafik-Library: Nutzen Sie die ganze Faszination der Vektor-Grafik. Mit dieser Bibliothek bauen Sie Vector-Grafiken auf einfache und schnelle Weise in Ihre Programme ein. Der Vorteil: Es wird nur wenig Speicherplatz und Rechenzeit verbraucht. Beschreibung auf Seite 22ff.

Zwei Disketten

Bestell-Nr. 45907

DM 34,90* sFr 29,50*/ öS 349,-*

Amiga-Sonderheft 2

Grafik, Anwendung

Object-Editor: Animierte Figuren, beispielsweise für eigene Spiele, entwickeln Sie mit diesem Editor auf komfortable Weise. Sogar mit Deluxe Paint erstellte Pinsel lassen sich einlesen.

Haushaltsbuch: Mit diesem hervorragenden Anwendungsprogramm verwalten Sie alle Einnahmen und Ausgaben auf übersichtliche Weise. Eine Monats- oder Jahresstatistik zeigt, in welchen Bereichen Sie zukünftig sparen können. Jetzt haben Sie Ihre Finanzen im Griff.

Keyboard-Master: Lernen Sie im Zehn-Finger-System zu tippen. Mit diesem didaktisch ausgereiften Programm ist dies kein Problem. Für Programmierer sind sogar Spezial-Lektionen mit wichtigen Sonderzeichen vorhanden.

FastLoadCopy: Dieses Tool bringt den DIR-Befehl auf Trab. Nach der »Operation« wird das Inhaltsverzeichnis einer Diskette im D-Zug-Tempo eingelesen. Zusätzlich kopiert das Programm Disketten und versieht diese mit dem schnellen Directory. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 2 mit einem Diskettensymbol gekennzeichnet sind.

3,5"-Diskette für Amiga

Bestell-Nr. 45802 **DM 29,90* sFr 24,90*/ öS 299,-***

Die Wiederbelebung für die

C64-Peripherie

Viele Amiga-Besitzer haben noch einen C64 mit Peripheriegeräten zu Hause stehen. Mit ein bißchen Hard- und Software können Sie diese zu neuem Leben erwecken und Ihre Daten so weiterbenutzen. Dabei ist die Bedienung wirklich einfach. Der fertig aufgebaute IEC-Handler erlaubt es, alte C64-Geräte wie die Floppy 1541 oder 1571, Commodore-MPS-Drucker und natürlich auch den C64 (zur Datenübertragung) am Amiga zu betreiben.

Das Gesamtpaket besteht aus der fertig aufgebauten Platine mit Verbindungskabel, der Treibersoftware auf 3,5"-Diskette sowie einer entsprechenden Dokumentation.

Bestell-Nr. 39101 **DM 79,-* sFr 71,-*/ öS 790,-***

Amiga-Sonderheft 8

Spiel Nr. 1: »Maze«

»Maze« ist das Spiel für Denker und Lenker. Schnelles Auffassungsvermögen und eine ausgeklügelte Taktik entscheiden dieses Spiel. Ein langes Spielvergnügen ist durch den eingebauten Level-Editor garantiert. Weitere Informationen über dieses faszinierende Spiel finden Sie auf Seite 106 in diesem Heft.

Spiel Nr. 2: »Oel-Fieber«

Steigen Sie ein in das Millionengeschäft um Öl und Aktien. Ihre Fähigkeiten als Manager werden auf immer neue Bewährungsproben gestellt. Das amüsante Würfelspiel erfordert dabei eine Menge Kreativität, um immer neue Strategien zu entwickeln. Wie Sie mit Öl- und Aktengeschäften reich werden können, lesen Sie in diesem Heft ab Seite 89.

Sicher ist sicher

Unumgänglich für jeden Amiga-User ist ein leistungsfähiges Anti-Viren-Programm. »Virus-Control v 1.4« schützt Sie vor allen bekannten Bootblock- und Linkviren sowie durch seine Lernfähigkeit auch vor zukünftigen »Infektionskrankheiten«. Schaffen Sie sich mit »Virus-Control v 1.4« ein zuverlässiges Immunsystem für Ihren Amiga. Die Anleitung finden Sie auf der Diskette.

Bestell-Nr. 45908

DM 29,90* sFr 24,90*/ öS 299,-*

* Unverbindliche Preisempfehlung

Wichtig:

Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,- können Sie Software-Disketten Ihrer Wahl aus dem Programmservice-Angebot im Wert von DM 180,- bestellen – egal ob diese DM 19,90, DM 29,90 oder DM 89,- kosten. Sie sparen DM 30,-!

Das Super-Software-Angebot finden Sie in den Zeitschriften

Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST-Magazin, PC Magazin, Computer live.

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken.

Das Scheckheft können Sie per Verrechnungsscheck oder mit der beigehefteten Zahlkarte direkt beim Verlag bestellen.

Kennwort: »Super-Software-Scheckheft« Bestell-Nr. W155

Impressum

Herausgeber: Carl-Frenz von Quadt, Otmar Weber
Redaktionsdirektor: Richard Kerler
Chefredakteur: Wolfram Höfler
Stellv. Chefredakteur: Gottfried Knechtel (km) – verantwortlich für den redaktionellen Teil
Chef vom Dienst: Susanne Kirmajer
Mitarbeiter dieser Ausgabe: René Beaupol, Uta Jürgens
Redaktionsassistent: Brigitte Bobensteller, Sylvia Derenthal, Helga Weber (202)
Telefax: 089/4613-778, Hotline: Donnerstag 14 bis 17 Uhr
 Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: Manuskripte und Programm Listings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für un-
 terlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Art-director: Friedemann Porscha
Titelgestaltung: Rolf Boyke
Layout: Alexander Kowarzyk (Cheflayout), Andrea Müller
Bildredaktion: Janos Feltser (Lig.), Sabine Tennstedt; Roland Müller (Fotografie), Ewald Standke, Norbert Raab (Spritzgrafik), Werner Nienstedt (Computergrafik)
Bildnachweis: Süddeutsche Zeitung (Titelmotiv)

Anzeigenleitung: Alicia Cleeß (313) – verantwortlich für die Anzeigen
Telefax: 089/4613-775

Anzeigenverwaltung und Disposition: Monika Burseg (147)

Auslandsrepräsentation:
Anzeigenverkaufsteilung Ausland: Thomas Schlüter (299)
Auslandsniederlassungen:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Fax 042-415770, Telex. 862329 mut ch
USA: M&T Publishing Inc., 501 Galveston Drive Redwood City, CA 94063, Telefon. (415) 366-3600, Telex 752-351
Österreich: Markt & Technik Ges. mbH, Große Neugasse 28, A 1040-Wien, Telefon. 0222/5871393, Telex 047-132532
Anzeigen-Auslandsvertretung:
England: F. A. Smyth & Associates Limited, 23a, Aymer Parade, London, N2 0PQ, Telefon 0044/13405058, Telefax. 0044/13419602
Israel: Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562255
Taiwan: Alm International Inc., 4F-1, No. 200, Sec. 2, Hsin-I Rd., Taipei, Taiwan, R.O.C. Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710
Korea: Young Media Inc., C.P.O. Box: 6113, Seoul/Korea, Tel. 0082-2-7564819, 1-7742759, Fax 0082-2-7575789

Vertriebsleitung: Helmut Grünfeldt (189)

Vertrieb Handelsaufträge: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Hauptstätter Straße 96, 7000 Stuttgart 1, Tel. 0711/6483-110

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-398. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 16,-

Produktion: Technik: Klaus Buck (Lig./180), Wolfgang Meyer (Stellv./867), Herstellung: Otto Albrecht (Lig./917)

Druck: SOV Graphische Betriebe, Leubanger 23, 8500 Bamberg

Urheberrecht: Alle in diesem Heft erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen, gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind

Haftung: Für den Fall, daß in diesem Heft unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-776

© 1990 Markt & Technik Verlag Aktiengesellschaft

Vorstand: Otmar Weber (Vors.), Bernd Balzer, Richard Kerler

Verlagsdirektor: Wolfram Höfler

Direktor Zeitschriften: Michael M. Pauly

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Telefon-Durchwahl im Verlag: Wählen Sie direkt. Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

```

60 xF '----- Interpreterschleife
61 xg OPEN pname$ FOR INPUT AS 1
62 RF WHILE NO$ EOF(1)
63 J02 LINE INPUT #1,a$-a$=UCASE$(LEFT$(a$,INSTR(a$+";",":")-1))
    zeile$a$
64 Tq WHILE zeile$a$ < > ""
65 634 a$=LEFT$(zeile$,INSTR(zeile$+"1","1")-1):zeile$=MID$(ze
    ile$,LEN a$+2)
66 g8 b$=LEFT$(a$,1) WHILE b$=" " OR b$=" " a$=MID$(a$,2)-b$:L
    EFT$(a$,1).WEND
67 oF befehl$a$ IF a$ < > "" THEN GOSUB ausführen
68 XL2 WEND
69 YMO WEND
70 7a CLOSE
71 J2 '----- Daten der Punkte umrechnen + speichern und Programme
    erzeugen
72 LQ GOSUB umrechnen:GOSUB linie.interpr
73 On OPEN pname$+".dat" FOR OUTPUT AS 1
74 JV FOR j=0 TO n
75 J22 FOR i=0 TO p:PRINT#1,MKI$(x(i,j));MKI$(y(i,j)), NEXT
76 J00 NEXT
77 Eh CLOSE
78 6w FOR i=1 TO LEN(zeichen0$)
79 1a2 a$=MID$(zeichen0$,i,1)
80 Qn a=INSTR(zeichen1$,a$)
81 QZ IF a THEN
82 9a4 IF a$="A" THEN
83 5I6 GOSUB Assembler.code
84 QZ4 ELSEIF a$="B" THEN
85 86 GOSUB Basic.code
86 Iq0 3 Hier können weitere Abfragen folgen.
87 QJ4 ELSE
88 MP END IF
89 NQ2 END IF
90 590 NEXT i
91 96 '----- Starten des erzeugten Basic-Programms oder Ende
92 Jd GOSUB schliessen
93 IS REM IF INSTR(zeichen1$,"B") THEN RUN pname$+".bas"
94 GB END
95 1T '***** Abbruch des Programms
96 qJ '----- Unterbrechung
97 mO unterbr:
98 PJ2 GOSUB schliessen
99 LGO END
100 ND '----- Fehlermeldung
101 9n f'
102 yk2 PRINT:PRINT"Fehler ".:BEEP
103 Mx COLOR 2,3:PRINT befehl$:COLOR 3,2
104 VP WHILE INKEY$="" WEND
105 Wq GOSUB schliessen
106 SMO END
107 9K ***** Unterprogramme (GOSUB)
108 f6 '----- Fenster + Screen schließen
109 4b schliessen.
110 ye2 WINDOW CLOSE 2
111 Ze SCREEN CLOSE 1
112 am0 RETURN
113 Wh '----- Befehl ausführen
114 3J ausführen.
115 Mb2 GOSUB untersuchen
116 xy PRINT " " b$ TAB(10) a$
117 zT b$=LEFT$(b$+";.....",5)
118 q8 c$= "DIM..PERS.SET..R0TI..ROT..LINE.MOVE.SHI..SHII.TURN.
    COPY.RES.."
119 ZM c$=c$+"ORIG.MASK.ACC..DEC..FORM.EXT..LET..BOX..LBOX.REC..
    LREC.INKS.."
120 XU a=(INSTR(c$,b$)+4)/5
121 EO IF a<1 OR a>1 AND flag1 THEN f
122 v1 IF RIGHT$(a$,1)="", THEN c$=a$+"0"
123 qF c$=a$
124 KW ON a GOSUB a.dim,c.pers,c.set,c.rot1,c.rot,c.line,c.move,
    c.sri,c.shii,c.turn,c.copy,c.rea,c.orig,c.mask,c.acc,c.acc
    o,form,c.ext,c.let,c.box,c.lbox,c.rec,c.lrec,c.inks
125 N20 RETURN
126 50 '----- a$ auf zulässige Zeichen prüfen
127 r9 untersuchen
128 22 b$="":c$=LEFT$(a$,1)
129 cT WHILE "A"<=c$ AND c$<="Z"
130 su4 b$=c$+c$ a$=MID$(a$,2)-c$:c$=LEFT$(a$,1)
131 YM2 WEND
132 yM IF b$=" " OR LEN(b$)>4 THEN f
133 JF c$=""
  
```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Fortsetzung)


```

134 gN IF a$ < > " THEN
135 lH4 c$ = ""
136 R5 FOR j=1 TO LEN(a$):c$=MID$(a$,j,1)
137 qU6 IF INSTR("XYZPMV:.,+-*/0123456789 ()",c$)=0 THEN f Z
LSE IF INSTR("(" )",c$)=0 THEN c$=c$+c$
138 J 4 NEXT
139 Et c$=c$
140 U2 END IF
141 drO RETURN
142 TD --- neues k berechnen
143 bc k
144 vo2 k=k+r a= k-u+1 OR k=-1
145 gE IF vo=2 AND a THEN k=k-r-r r=-r
146 na IF mo=1 AND a THEN k=1
147 jLc RETURN
148 4r --- x0,y0,z0 holen
149 Xm x0y0z0
150 E52 flag4=-1:GOSUB h:x0=a:GOSUB h:y0=a:GOSUB h:z0=a
151 nPO RETURN
152 Q3 --- addieren bei DEC,ACC,SHI,SHL
153 LO add
154 Br2 IF xx THEN x(1,j)=x(1,j)+x0*a
155 XD IF yy THEN y(1,j)=y(1,j)+y0*a
156 tZ IF zz THEN z(1,j)=z(1,j)+z0*a
157 tVO RETURN
158 Jt '----- Parameter von c$ nach a holen
159 qJ n
160 XB2 s=0:s2=0 f$=""
161 EY d=INSTR(c$,""):IF d THEN f$=","
162 55 a=INSTR(c$,""):IF a AND (a<d OR d=0) THEN f$=" " d=s
163 Jd a=INSTR(c$,".."):IF a AND (a<d OR d=0) THEN f$=".." d=s
164 c$ IF d THEN d$=" "+LEFT$(c$,d-1):c$=MID$(c$,d+LEN(f$)) ELSE
d$=" "+c$ c$=""
165 yv IF d$ < > " THEN
166 2I WHILE d$ < > ""
167 Qq4 f$=LEFT$(d$,1):GOSUB objekt.holen
168 tN IF f$="+" THEN
169 d76 s=s+s2 s2=s
170 UN4 ELSEIF f$="-" THEN
171 8p6 s=s-s2 s2=-s
172 ND4 ELSEIF f$="*" THEN
173 7b6 s2=s2*a
174 e24 ELSEIF f$="/" THEN
175 wT6 IF a=0 THEN f ELSE s2=s2/a
176 rV4 ELSEIF f$="\" THEN
177 Kx6 IF a=0 THEN f ELSE s2=s2\ a
178 TC4 ELSE
179 pw6 GOTO f
180 qJ4 END IF
181 MA2 WEND
182 s1 END IF
183 JH a=s-s2
184 KwO RETURN
185 lT objekt.holen.
186 UO2 d$=MID$(d$,2) e$=LEFT$(d$,1):IF e$=" THEN f
187 IQ v=1
188 wc WHILE e$="+" OR e$="-"
189 lJ4 IF e$=" THEN v=-v
190 M9 d$=v.d$ d$,2 e$=LEFT$(d$,1)
191 WK2 WEND
192 z4 IF INSTR("0123456789",e$) THEN 'Zahl
193 gB4 GOSUB ziffern.holen:a=d
194 vG2 ELSEIF e$="N" THEN 'Anzahl der Bilder
195 RB4 a=n+1 d$=MID$(d$,2)
196 v2 ELSEIF e$="P" THEN 'Anzahl der Punkte
197 nM4 a=p+1:d$=MID$(d$,2)
198 oM2 ELSEIF INSTR("XYZ",e$) THEN 'Koordinate
199 QQ4 g$=e$ d$=MID$(d$,2):GOSUB ziffern.holen
200 O7 IF g$="X" THEN a=x(d,an) ELSE IF g$="Y" THEN a=y(d,an) ELSE
a=z(d,an)
201 -Q2 ELSEIF e$="V" THEN 'variable
202 vP4 d$=MID$(d$,2):GOSUB ziffern.holen:a=v d
203 sb2 ELSE
204 EL4 GOTO f
205 F62 END IF
206 TL a=v*a
207 hJO RETURN
208 Xg zahl.holen
209 yL2 a=1
210 y WHILE e$="+" OR e$="-"
211 gL4 IF e$=" THEN a=-a
212 iV d$=MID$(c$,2) e$=LEFT$(c$,1)
213 eg2 WEND
214 D8 GOSUB ziffern.holen a=a*d

```

```

215 pRO RETURN
216 Se ziffern.holen
217 2a2 komma=0:e$=" " h2=LEFT$(d$,1):IF h2="OR INSTR("0123456789",h2)=0 THEN f
218 Zw WHILE INSTR("0123456789",h2) AND d$ < > ""
219 e44 IF n$=" THEN IF komma THEN f ELSE komma=1
220 st e$=e$+n$:d$=MID$(d$,2):h2=LEFT$(d$,1)
221 Da2 WEND
222 Ax d=VAL(e$)
223 x/c T N
224 ZF --- (i,j) -er Punkt im wx und wz drehen
225 JQ drehen
226 -J2 IF z THEN
227 RQ4 d1=x(1,j)-x0:d2=y(1,j) y0:GOSUB winkel:w=w+wz*v
228 SH x(1,j)=x0+COS(w)*b
229 T2 y(1,j)=y0+SIN(w)*b
230 AX END IF
231 x IF x THEN
232 bV4 d1=x(1,j)-x0:d2=y(1,j) y0:GOSUB winkel:w=w+wx*v
233 L6 x(1,j)=x0+COS(w)*b
234 MZ y(1,j)=y0+SIN(w)*b
235 J2 END IF
236 EN RETURN
237 S --- Parameter für Rotationsbefehle holen
238 Fq get.rot
239 WK2 GOSUB x0y0z0:GOSUB h:wx=a*grad:GOSUB h wz=a*grad
240 Xu flag6=c$ < > " :GOSUB h:dw=a*grad:flag7=c$ < > ""
241 9q IF flag7 THEN
242 114 x1=x grad y1=y grad z1=z dw/grad
243 91 GOSUB dw=a*grad
244 s12 END IF
245 JvO RETURN
246 M6 '----- JD- in 2D-Koordinaten umrechnen
247 9z umrechner
248 12 x1= x/c.gx fx y1= y/c.orgy /y zf=(zf+orgz)/fz
249 BK FOR j=0 TO n
250 9R4 FOR i=0 TO p
251 VV6 e=(z(1,j)+orgz)/fz
252 8g x(1,j)= (a*x1+(zf-a)*(x(1,j)+orgx)/fx)/zf
253 a3 y(1,j)=255-(a*y1+(zf-a)*(y(1,j)+orgy)/fy)/zf/2
254 BG4 NEXT
255 QM2 NEXT
256 V6O RETURN
257 wZ '----- Zingsbe: d1,d2 = Koord. e. Vektors
258 - Ausgabe: w = Zwischenwinkel, b = Betrag
259 zE -irke
260 gn2 IF d1=0 THEN
261 oT4 w=pi/2:IF d2<0 THEN w=w*3
262 pY2 ELSE
263 tB4 w=ATN(d2/ABS(d1)):IF d1<0 THEN w=pi-w
264 C52 END IF
265 SN b=SQR(d1*d1+d2*d2)
266 eGO RETURN
267 - Eingabe: x0,y0,z0, x1,y1,z1
268 eJ Ausgabe: wx,wz
269 NQ trans.
270 j32 IF x1-x0 OR y1-y0 OR z1-z0 THEN ELSE f
271 l8 d1=x1-x0:d2=y1-y0:GOSUB winkel:w=w+pi/2
272 6L d1=x1-x0:d2= b:GOSUB winkel:wx=w+pi/2
273 J2 WEND
274 - linie$ interpretieren
275 OX linie.interpr
276 st2 a$=MID$(linie$,2) linie 1)=1 j=2 'Move
277 f - linie e$ < > ""
278 -4 a=LEFT$(a$,"") c$=LEFT$(a$,a-1) a$=MID$(a$,a+1)
279 Qs IF RIGHT$(c$,1)=", " THEN c$=c$+"0"
280 p4 WHILE c$ < > ""
281 m6 GOSUB intervall.holen
282 h7 IF d=0 THEN m2=m1
283 B4 FOR i=m1 TO m2 STEP FNst(m1,m2)
284 Cp8 linie(j)=i:linie(j+1)=2:j=j+2 'Draw
285 gL6 NEXT
286 3r4 'Draw
287 zS linie(j-1)=1 'doch Move
288 522 WEND
289 f 'linie C = j
290 2eO RETURN
291 3o intervall.holen:
292 mU2 a=INSTR(c$,".."):b=INSTR(c$,"")
293 mJ IF a=0 OR a>b AND b THEN
294 1J4 GOSUB n wias d c

```



```

295 M52 ELSE
296 lb4 IF a=1 THEN m1=0 c$=MID$(c$,3) ELSE GOSUB a m1 a
297 fj IF LEFT$(c$,1)=", " OR c$="" THEN m2=p.c$=MID$(c$,2) ELSE G
GOSUB h:m2=a
298 ef d=-1
299 le2 END IF
300 GoQ RETURN
301 bq ***** Unterprogramme für die einzelnen Befehle.
302 M7 c.dim
303 qm2 flag1=NOT flag1:IF flag1 THEN f
304 7b GOSUB h:p=a-1:GOSUB h:n=a-1:flag3=c$=""
305 pb DIM x(p,n),y(p,n),z(p,n),punkt(p)
306 BR FOR i=0 TO p:punkt(i)=-1:NEXT
307 m1 an=0:en=n:en2=n:mo=0
308 KwQ RETURN
309 Qb c.pers:
310 wq2 GOSUB h:xf=a:GOSUB h:yf=a:GOSUB h:zf=a
311 NaQ RETURN
312 lW c.set:
313 xp2 GOSUB h:set:a=INSTR(c$+" ", " ")
314 2B b$=MID$(c$,a+1) c$=LEFT$(c$,a-1)
315 Qd WHILE c$<>""
316 QH4 GOSUB x0y0z0:a=0:m2=n
317 Hq IF c$<>"" THEN GOSUB h:m2=a:GOSUB h
318 fh FOR i=a TO m2 STEP FNst(a,m2)
319 ka6 IF xx THEN x(set,i)=x0
320 IF yy THEN y(set,i)=y0
321 IV IF zz THEN z(set,i)=z0
322 HMA NEXT
323 ex set=set+1:a=INSTR(b$+" ", " ")
324 lY c$=MID$(b$,a+1) b$=MID$(b$,a-1)
325 GU2 WEND
326 eED RETURN
327 G4 c.move:
328 f22 ERASE punkt:DIM punkt(p)
329 AS a=INSTR(c$," ") IF a<2 THEN f
330 R b$=MID$(c$,a+1) c$=LEFT$(c$,a-1)
331 G1 IF RIGHT$(c$,1)=", " THEN c$=c$+"0"
332 fu WHILE c$<>""
333 eb4 GOSUB intervall.holen
334 KJ IF d=0 THEN m2=m1
335 w2 FOR i=m1 TO m2 STEP FNst(m1,m2):punkt(i)=i:NEXT
336 rf2 WEND
337 ex c$=b$:GOSUB h:an=a
338 jf en=an:IF c$<>"" THEN GOSUB h:en=a
339 gw en2=en:IF c$<>"" THEN GOSUB h:en2=a
340 y5 GOSUB h:mo=a
341 lD IF en<an OR en2<an OR ABS(mo-1)>1 THEN f
342 n1 IF en2=en THEN mo=0
343 tVQ RETURN
344 DQ c.line:
345 KR2 linie$=linie$+" "+a$
346 wYQ RETURN
347 4y c.rot:
348 uR2 GOSUB get.rot
349 on IF flag7 THEN GOSUB trans
350 u maen=an+1
351 m4 FOR i=0 TO p
352 c4 IF punkt(i) THEN
353 fE6 k=i:r=1
354 nd FOR e=an TO en2:j=FNnd(e)
355 dy8 v=-1:GOSUB drehen
356 c6 d1=x(1,j)-x0:d2=z(1,j)-z0
357 l1 GOSUB winkel:w=w+dw/n$K
358 M7 x(1,j)=x0+(cos w)*b
359 Tn z(1,j)=z0+(sin w)*b
360 c1 v=1:GOSUB drehen:GOSUB k
361 w6 NEXT
362 m4 END IF
363 w12 NEXT
364 E4Q RETURN
365 B5 c.rot1.
366 wt2 Push en:Push mo
367 Kt4 en=an:mo=1:GOSUB c.rot
368 J42 Pop mo:Pop en
369 JvQ RETURN
370 nD c.turn:
371 Hd2 GOSUB get.rot
372 cM IF flag6 THEN x1=wx/grad:y1=wz/grad:z1=dw/grad:GOSUB tran
s
373 BQ FOR i=0 TO p
374 y14 IF punkt(i) THEN
375 b16 FOR e=an TO en2
376 e18 j=FNnd(e) v=1:GOSUB drehen
377 AP6 NEXT
378 2v4 END IF
379 GB2 NEXT
380 U6Q RETURN
381 l7 c.sh1.
382 eF2 GOSUB x0y0z0:m=en-an+1
383 la FOR i=0 TO p
384 f4 IF punkt(i) THEN
385 z6 k=i:r=1
386 t1 FOR e=an TO en2
387 f8 j=FNnd(e):a=k/n:GOSUB add:GOSUB k
388 z6 NEXT
389 e4 END IF
390 G2 NEXT
391 tVQ RETURN
392 Gh c.sc11
393 x2 Push en:Push mo
394 z4 en=an:mo=1:GOSUB c.sh1
395 w2 Pop mo:Pop en
396 zMQ RETURN
397 f c.cop.
398 s2 flag4=-1
399 lx GOSUB h:m1=a:GOSUB h:m2=n
400 Gh IF c$<>"" THEN GOSUB n m3=a ELSE m3=m2:m2=m1+FNst(m1,m2)
401 s IF c$<>"" THEN
402 T4 GOSUB h:b=FNst(m1,m2):c=FNst(m3,a)
403 l IF ABS((m2-m1)/(a-m3))<>1 THEN f
404 v FOR i=0 TO p
405 f6 IF punkt(i) THEN
406 y8 FOR e=m1 TO m2 STEP b
407 z4 j=FNnd(e)
408 z1 k=FNnd(e-m1)/b*c+m3)
409 f IF xx THEN x(1,k)=x(1,j)
410 sa IF yy THEN y(1,k)=y(1,j)
411 t IF zz THEN z(1,k)=z(1,j)
412 f8 NEXT
413 f6 END IF
414 z6 NEXT
415 l2 ELSE
416 p74 FOR i=0 TO p
417 f6 IF punkt(i) THEN
418 w8 FOR e=m2 TO m3 STEP FNst(m2,m3)
419 wSA j=FNnd(e)
420 z1 IF xx THEN x(1,j)=x(1,m1)
421 w IF yy THEN y(1,j)=y(1,m1)
422 f IF zz THEN z(1,j)=z(1,m1)
423 z8 NEXT
424 f6 END IF
425 w14 NEXT
426 c2 END IF
427 tVQ RETURN
428 TH c.res
429 VR2 x0=fx:y0=fy:z0=fz
430 CC fx=1:IF c$<>"" THEN GOSUB h fx=a/640
431 l1 fy=fx:IF c$<>"" THEN GOSUB h fy=a/512
432 2U fz=fx:IF c$<>"" THEN GOSUB h fz=a/640
433 Xv IF fx*fy*fz=0 THEN f
434 w2 x=x/x0*fx:y=y/y0*fy:z=z/z0*fz
435 AS orgx=orgx/x0*fx:orgy=orgy/y0*fy:orgz=orgz/z0*fz
436 Gk IF flag4 THEN
437 AS4 FOR i=0 TO p
438 B46 FOR j=0 TO n
439 7a8 x(1,j)=x(1,j)/x0*fx
440 P3 y(1,j)=y(1,j)/y0*fy
441 hK z(1,j)=z(1,j)/z0*fz
442 D16 NEXT
443 EJ4 NEXT
444 e2 END IF
445 l1 flag4=-1
446 Y4Q RETURN
447 f4 flag
448 c2 x=orgx:y0=orgy:z0=orgz
449 yD GOSUB h:orgx=a:GOSUB h:orgy=a:GOSUB h:orgz=a
450 Ky IF flag4 THEN
451 Og4 FOR i=0 TO p
452 P66 FOR j=0 TO n
453 y8 x(1,j)=x(1,j)+x0-orgx
454 D3 y(1,j)=y(1,j)+y0-orgy
455 Sx z(1,j)=z(1,j)+z0-orgz
456 R46 NEXT
457 SX4 NEXT
458 ED2 END IF
459 v xf=xf+x0-orgx:yf=yf+y0-orgy:zf=zf+z0-orgz
460 M0Q RETURN

```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Fortsetzung)


```

461 C4 c.mask
462 DN2 GOSUB h:xx=a GOSUB h:yy=a GOSUB n zz=a
463 pRO RETURN
464 Pm o.aco:
465 UH2 flag2=LEFT$(b$,3)="DEC"
466 6b GOSUB x0y0z0:x=en-an+1
467 ew Fm 1 0 TO p
468 144 IF punkt(1) THEN
469 Q16 k=1 IF flag2 THEN k=m-1
470 dF FOR e=en TO en2
471 118 j=FNmd(e)
472 7Q a=k*x/m/π k=k+1
473 1L IF flag2 THEN a=1-a:k=k-2
474 1A GOSUB add
475 kp6 NEXT
476 oV4 END IF
477 mx2 NEXT
478 4gD RETURN
479 8D c.form
480 112 GOSUB h:m1=a:GOSUB h:m2=a:ABS(m2-m1)
481 sA FOR i=0 TO p
482 114 IF punkt(1) THEN
483 vH6 x0=x(1,m2)-x(1,m1)
484 gv y0=y(1,m2)-y(1,m1)
485 x9 z0=z(1,m2)-z(1,m1)
486 Bv a=FNst(m1,m2):k=1
487 vF FOR e=m1+a TO m2-a STEP a:j=FNmd(e)
488 dQ8 IF xx THEN x(1,j)=x(1,m1)+x0*k/m
489 OI IF yy THEN y(1,j)=y(1,m1)+y0*k/m
490 N8 IF zz THEN z(1,j)=z(1,m1)+z0*k/m
491 75 k=k+1
492 166 NEXT
493 tm4 END IF
494 382 NEXT
495 LxO RETURN
496 M8 c.ext:
497 OR2 a$=c$:GOSUB x0y0z0:GOSUB h:GOSUB h:Push an:Push en2
498 JQ IF c$<>" THEN GOSUB h:an=a:GOSUB h:en2=m2
499 J c$=a$
500 e1 GOSUB x0y0z0:GOSUB h:m1=a:GOSUB h:m2=a
501 JR m3=an,m4=en2 IF c$<>" THEN GOSUB h:m3=a:GOSUB n:m4=a IF
m4<m3 THEN f
502 DV FOR i=0 TO p
503 3d4 IF punkt(1) THEN
504 En6 FOR e=m3 TO m4:j=FNmd(e)
505 6k8 a=m1+m2-m1/(m4-m3)*(e-m3)
506 lo IF xx THEN x(1,j)=x0+(x(1,j)-x0)*a
507 CE IF yy THEN y(1,j)=y0+(y(1,j)-y0)*a
508 ge IF zz THEN z(1,j)=z0+(z(1,j)-z0)*a
509 IN6 NEXT
510 AJ4 END IF
511 KP2 NEXT
512 7F Pop en2:Pop an
513 dFO RETURN
514 2D c.let:
515 Xn2 IF LEFT$(c$,1)<>" THEN f
516 Ry a=INSTR c$+"," ",") d$=MID$(c$,2,a-2):c$=MID$(c$,a+1)
517 vL GOSUB zlferrn.holen:m1=d:GOSUB h:v(m1)=a
518 1KO RETURN
519 8a c.box
520 Y2 GOSUB daten.holen:IF p(0)=8 THEN f
521 1 FOR e=m1 TO m2:j=FNmd(e)
522 w4 a=1:x1=x0:y1=y0:z1=z0:GOSUB s
523 3b x1=x0+dx:GOSUB s
524 02 z1=z0+dz:GOSUB s
525 6k x1=x0:GOSUB s
526 wp y1=y0+dy:z1=z0:GOSUB s
527 nA x1=x0+dx:GOSUB s
528 d4 z1=z0+dz:GOSUB s
529 Ao x1=x0:GOSUB s
530 d12 NEXT
531 vXO RETURN
532 WH s
533 Lx2 IF xx THEN x(p(a),j)=x1
534 dE IF yy THEN y(p(a),j)=y1
535 vI IF zz THEN z(p(a),j)=z1
536 aW a=a+1
537 180 RETURN
538 Ar dater.holen
539 QB2 ERASE p DIM p(5)
540 Zr a=INSTR(c$," ") IF a<2 THEN f
541 nQ b$=MID$(c$,a+1) c$=LEFT$(c$,a-1)
542 r7 IF RIGHT$(c$,1)=", THEN c$=c$+"0"
543 4J WHILE c$<>"

```

```

544 104 GOSUB intervall.holen
545 w1 IF d=0 THEN m2=m1
546 rD FOR i=m1 TO m2 STEP FNst(m1,m2):p(0)=p(0)+1:p(p(0))+1:N
EXT
547 C42 WEND
548 mC c$=b$
549 L2 GOSUB x0y0z0:GOSUB h:dx=a:dy=a:dz=a:m1=0:m2=n
550 40 IF c$<>" THEN GOSUB h:dy=a
551 c7 IF c$<>" THEN GOSUB h:dz=a
552 Kn IF c$<>" THEN GOSUB h:m1=a
553 K8 IF c$<>" THEN GOSUB h:m2=a IF m2<m1 THEN f
554 110 RETURN
555 16 c.box
556 J82 GOSUB c.box
557 GX c$="123456785:26:37:48":GOSUB anhaengen
558 MyO RETURN
559 Cp anhaengen:
560 Sb2 linie$=linie$+": "
561 H7 FOR i=1 TO LEN(c$)
562 e54 a=VAL(MID$(c$,i,1))
563 CL IF a THEN
564 N16 linie$=linie$+MID$(STR$(p(a)),2)+", "
565 1R4 ELSE
566 HD6 MID$(linie$,LEN(linie$),1)=" "
567 5Y4 END IF
568 FK2 NEXT
569 4k linie$=LEFT$(linie$,LEN(linie$)-1)
570 YAO RETURN
571 E4 c.rec:
572 r92 GOSUB daten.holen IF p(0)=4 OR dx*dy*dz THEN f
573 3Y FOR e=m1 TO m2:j=FNmd(e)
574 bm4 a=1:x1=x0:y1=y0:z1=z0:GOSUB s
575 5U x1=x0+dx:IF dx=0 THEN z1=z0+dz
576 n GOSUB s
577 Kg y1=y0+dy:IF dy THEN z1=z0+dz
578 jw GOSUB s
579 U1 x1=x0:IF dx=0 THEN z1=z0
580 ly GOSUB s
581 XA2 NEXT
582 KNO RETURN
583 v1 c.lrec
584 PX2 GOSUB c.rec
585 1A c$="12341" GOSUB anhaenger
586 cQO RETURN
587 JA c.links
588 Q12 FOR i=1 TO 0 STEP-1
589 nF4 GOSUB x0y0z0
590 L2 ink 0+1*3:x0 ink(1+1*3)=y0 ink(2+1*3)=z0
591 en2 NEXT
592 uW0 RETURN
593 d9 '----- Basicprogram auf Diskette erzeugen
594 1m basic.code
595 nY Basic.code.data:RESTORE Basic.code.data
596 nt2 OPEN pname$+".bas" FOR OUTPUT AS 1
597 gq PRINT #1,"LIBRARY"+CHR$(34)+"intuition.library"
598 Ja PRINT #1,"LIBRARY"+CHR$(34)+"graphics.library"
599 bn PRINT #1,"CLEAR,100000"
600 1N Read.data
601 t DATA "SCREEN 2,640,256,1,2","WINDOW 2,,,0,2"
602 3y DATA scr2$=PEEK(L(WINDOW(7)+46),bm2$=PEEK(L(WINDOW(8)+4),#
603 54 PRINT #1,"PALETTE 0,"+STR$(ink(0)/15)+","+STR$(ink(1)/15)
+","+STR$(ink(2)/15)
604 Sb PRINT #1,"PALETTE 1,"+STR$(ink(3)/15)+","+STR$(ink(4)/15)
+","+STR$(ink(5)/15)
605 K8 Read.data
606 Qn DATA "SCREEN 1,640,256,1,2","WINDOW 2,,,0,1"
607 TX DATA rp$=WINDOW(8),bm1$=PEEK(rp$+4),#
608 wT PRINT #1,"OPEN"+CHR$(34)+pname$+".dat"+CHR$(34)+"FOR INPU
T AS 1"
609 OE PRINT #1,"p$="+STR$(p)+" n$="+STR$(n)+" DIM x$(p,n),y$(
p,n)"
610 Ju PRINT #1,"x$="+STR$(x$)+":y$="+STR$(y$)+":z$="+STR$(z$)+":CVI(INPUT
611 TJ PRINT #1,"FOR j$=0 TO n$:FOR i$=0 TO p$:x$(i$,j$)=CVI(
3(2,1)) :y$(i$,j$)=CVI(3(2,1)) :z$(i$,j$)=CVI(3(2,1)) NEXT NEXT"
612 1D PRINT #1,"j$=0:ScreenToFont scr2$:POKEW &HDF0+256+&H96,
32:POKEW &HDF0+256+&H144,0"
613 Un PRINT #1,"WHILE PEEK(-4202495) AND 64"
614 x6 PRINT #1," CLS"
615 J8 IF flag3 THEN PRINT #1," Move rp$,0,251:Draw rp$,x$,y$
:Draw rp$,631,251"
616 qQ FOR i=1 TO linie(0)STEP 2

```

Listing. Dreidimensionale Animationen mit »3DAnim«
(Fortsetzung)


```

617 s54 a$=MID$(STR$(linie(ii+1)),2)
618 Bq b$="Draw":IF linie(ii)=1 THEN b$="Move"
619 vp PRINT #1," "+b$+" r",x,x,"+a$+",j%,j% "-a$-",j%
620 s52 NEXT
621 O1 Read.data
622 y8 DATA " BitBitmap bml,0,0,bm2,0,0,640,256,EHC0,-1,0"
623 LC DATA " i%=(j%-1)MOD(n%-1)
624 wF DATA WEND,WINDOW CLOSE 2,SCREEN CLOSE 1,WINDOW CLOSE 2,SC
PEEN CLOSE 2
625 Rh DATA LIBRARY CLOSE,*
626 yF CLOSE
627 T50 RETURN
628 ks '----- Assemblerprogramm auf Diskette erzeugen
629 RL Assembler.code:
630 3c Assembler.code.data:RESTORE Assembler.code.data
631 GW2 OPEN pname$+".asm"FOR OUTPUT AS 1
632 EG PRINT #1," bra run"
633 mA PRINT #1,"intname: dc.b 'intuition.library',0"
634 14 PRINT #1,"gfxname: dc.b 'graphics.library'"
635 35 PRINT #1,"title dc.b 'C'"
636 1n PRINT #1,"dosname: dc.b 'dos.library',C"
637 Gy Read.data
638 Mu DATA "defscr: dc.w 0,0,640,256,1,1,$8000,15"," dc.l 0,ait
le,0,0"
639 pk DATA "defwin: dc.w 0,0,640,256,1"," dc.l 0,$800,0,0,title
"
640 95 DATA "screen1: dc.l 0,0"," dc.w 0,0,640,256,15","screen2:
dc.l 0"
641 HH DATA "window1: dc.l 0","rastport: dc.l 0","bitmap1: dc.l
0"
642 R3 DATA "window2: dc.l 0","bitmap2: dc.l 0","adr: dc.l 0","
643 Sn PRINT #1,"filename: dc.b '"+pname$+".dat',0"
644 N5 Read.data
645 O1 DATA " even","filehd: dc.l 0","bild: dc.w 0","run: move.l
4,a6",*
646 qZ PRINT #1,"p: equ "+STR$(p+1)
647 XM PRINT #1,"n equ "+STR$(n+1)
648 J1 PRINT #1,"xf: equ "+STR$(xf)
649 40 PRINT #1,"yf: equ "+STR$(255-yf/2)
650 TB Read.data
651 uy DATA " lea dosname,a1"," jsr -408(a6)"," move.l d0,a3","
move.l #filename,d1"
652 Gp DATA " move.l #1005,d2"," jsr -30(a3)"," move.l d0,fileh
d"," bne w2"
653 OW DATA " move.l a3,a1"," jmp -414(a6)"," w2: move l #n*p*4,
d0"," moveq #2,d1"
654 Fx DATA " jsr -198(a6)"," move.l d0,adr"," move.l d0,d2"," m
ove.l filehd,d1"
655 pO DATA " move.l #n*p*4,d3"," jsr -42(a3)"," move.l d0,d6","
" move.l filehd,d1"
656 59 DATA " jsr -36(a3)"," move.l a3,a1"," jsr -414(a6)"
657 12 DATA " lea gfxname,a1"," jsr -408(a6)"," move.l d0,a1"
658 5W DATA " lea intname,a1"," jsr -408(a6)"," move.l d0,a1"
659 N3 DATA " move.l d0,a6"," lea defscr,a0"," jsr -198(a6)"
660 au DATA " move.l d0,screen1"," lea defwin,a0"," jsr -204(a6)
"
661 LN DATA " move.l d0>window1"," move.l d0,a0"," move.l 50(a0)
,a0"
662 Ot DATA " move.l a0,rastport"," move.l 4(a0),bitmap1"
663 tO DATA " lea defscr,a0"," jsr -198(a6)"," move.l d0,screen2
"
664 yH DATA " add.l #184,d0"," move.l d0,bitmap2"
665 eb DATA " sub.l #140,d0"," move.l d0,a0"," moveq #0,d0","
666 wE PRINT #1," moveq #"+MID$(STR$(ink(0)),2)+"",d1"
667 7Y PRINT #1," moveq #"+MID$(STR$(ink(1)),2)+"",d2"
668 11 PRINT #1," moveq #"+MID$(STR$(ink(2)),2)+"",d3"
669 mU Read.data
670 Ox DATA " move.l a3,a6"," move.l a0,-[sp]"
671 2p DATA " jsr -288(a6)"," move.l (sp)+,a0"," moveq #1,d0","
672 HJ PRINT #1," moveq #"+MID$(STR$(ink(3)),2)+"",d1"
673 Sw PRINT #1," moveq #"+MID$(STR$(ink(4)),2)+"",d2"
674 d9 PRINT #1," moveq #"+MID$(STR$(ink(5)),2)+"",d3"
675 sa Read.data
676 w6 DATA " jsr -288(a6)"," move #n,bild"," move #32,$dff096
"," clr.l $dff144"
677 w1 DATA " move.l a3,a6"," move.l adr,a5"," loop: move.l rastp
ort,d7"
678 1a DATA " move.l d7,a1"," moveq #0,d0"," moveq #0,d1"," js
r -240(a6)"
679 R8 DATA " move.l d7,a1"," jsr -48(a6)","
680 3n IF flag3 THEN CALL Read.data
681 zP DATA " move.l d7,a1"," moveq #0,d0"," move #255,d1","

```

```

sr -24(a6)"
682 LE DATA " move.l d7,a1"," move #xf,d0"," move #yf,d1"," js
r -246(a6)"
683 GK DATA " move.l d7,a1"," move #639,d0"," move #255,d1","
jsr -246(a6)","
684 wF FOR ii=1 TO linie(0)STEP 2
685 wV4 PRINT #1," move.l d7,a1"
686 kK a=4*linie(ii+1):IF a THEN a$=STR$(a)ELSE a$=" "
687 yK PRINT #1," move "+a$+"(a5),d0"
688 x3 PRINT #1," move "+STR$(a+2)+"(a5),d1"
689 30 PRINT #1," jsr "+STR$(234-6*linie(ii))+"(a6)"
690 D12 NEXT
691 sLO datas1:RESTORE datas1
692 3-2 Read.data
693 OR DATA " move.l bitmap1,a0"," moveq #0,d0"," moveq #0,d1"
694 Sx DATA " move.l bitmap2,a1"," moveq #0,d2"," moveq #0,d3"
695 wQ DATA " move #640,d4"," move #256,d5"," moveq #-340,d6"
696 QT DATA " moveq #-1,d7"," move.l d0,a2"," jsr -30(a6)"
697 6d DATA " move bild,d0"," subq #1,d0"," bne w1"," move #n,
d0"
698 JR DATA " sub.l #n*p*4,a5"," w1: move d0,bild"," add.l #p*4
,a5"
699 FV DATA " btst #6,$bfe001"," bne loop"
700 B1 DATA " move.l a4,a6"," move.l window1,a0"," jsr -72(a6)"
701 OA DATA " move.l screen2,a0"," jsr -66(a6)"," move.l screen1
,a0"," jsr -66(a6)"
702 9a DATA " move.l 4,a6"," move.l adr,a1"," move.l #n*p*4,d0"
"," jsr -210(a6)"
703 ge DATA " move.l a4,a1"," jsr -414(a6)"," move.l a3,a1"," jm
p -414(a6)"," end",*
704 1a CLOSE
705 JLO RETURN
706 1K '***** Unterprogramme (SUB) :
707 LR '----- x auf dem Stapel merken
708 rW SUB Push(x)STATIC
709 602 stapel$=STR$(x)+CHR$(0)+stapel$
710 UMO END SUB
711 7R '----- x vom Stapel holen
712 1d SUB Pop(x)STATIC
713 2J2 x=VAL(stapel$).stapel$=MID$(stapel$,INSTR(stapel$,CHR$(0)
)-1)
714 Ya0 END SUB
715 8R '----- DATAs bis "a" lesen und auf Disk schreiben
716 sk SUB Read.data STATIC
717 CP2 READ a$:WHILE a$<>"*.PRINT #1,a$:READ a$:WEND
718 ce0 END SUB

```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Schluß)

```

DIM 54,144
ORIG 50,100,0

SET 0,100,150:150,250:200,150:125,200:175,200 ;A
SET 5,210,150:210,250:240,220:270,250:270,150 ;M
SET 10,280,150:280,250 ;I
SET 12,350,250:290,250:290,150:350,150:350,200:320,200 ;G
SET 18,360,150:410,250:460,150:385,200:435,200 ;A
SET 23,10,20:10,120:40,90:70,120:70,20 ;M
SET 28,80,20:130,120:180,20:105,70 155,70 ;A
SET 33,240,120:190,120:190,20:240,20:240,70:210,70 ;G
SET 39,250,20:300,120:350,20:275,70 325,70 ;A
SET 44,360,120:410,120:360,20:410,20 ;Z
SET 48,420,20:420,120 ;I
SET 50,430,20:430,120:480,20:480,120 ;N

LINE 0,1,2:3,4:5..9:10,11:12..17:18..20:21,22
LINE 23..27:28..30:31,32:33..38:39..41:42,43:44..47:48,49:50..53

MOVE 0..22:0,n/2-1,n-1,2ROT 300,100,0,300,100,10,360

ROT 300,100,0,300,100,10,360

MOVE 23..p-1:0,n/2-1,n-1,2

FOR 30,100,0,300,100,10,360

```

Listing 2. Ein Beispiel für die Fähigkeiten von »3DAnim« liefern diese Befehle

Nur Fliegen

Ein Flugsimulator zum Abtippen: Das gibt es nur im AMIGA-Magazin und im AMIGA-Sonderheft. Zumindest in dieser hervorragenden Geschwindigkeit, tollen Grafik und der Kürze des Listings.

von Michael Koch

Wolten Sie schon immer einmal fliegen? Die billigste und ungefährlichste Lösung ist ein Flugsimulator. Es ist schon erstaunlich, daß ein solches Programm in einer Hochsprache auf dem Amiga so kurz und leistungsfähig sein kann. Nicht nur die Landebahn, sondern Funktürme, ein Wald, ein See mit Insel, sich drehende Windräder und Vögel sind in »Flusi« realisiert. Erweiterungen um andere Landschaftselemente stellen für den geübten C-Programmierer kein Problem dar.

Nach dem Abtippen und Übersetzen starten Sie das Programm vom CLI aus mit folgendem Aufruf:

Flusi

Schnell ist der Bildschirmaufbau beendet, und Sie können losfliegen. Aber halt, erst ein paar Erläuterungen zur Steuerung. Die Instrumente sind im unteren Drittel angeordnet (Bild 1, Seite 41 oben; und Bild 2). Der Geschwindigkeitsmesser zeigt die Flugeschwindigkeit in Knoten an. Bei zirka 50 Knoten können Sie abheben. Auf dem Kompaß sehen Sie die aktuelle Flugrichtung. Der Drehzahlmesser zeigt die Umdrehungszahl des Motors an. Die Instrumente Quer-, Seiten- und Höhenruder teilen Ihnen die aktuelle Stellung der Ruder mit. Achtung, sie schwenken nicht automatisch in die Neutralstellung zurück. Beim Höhenmesser ist der kleine Zeiger für die Zehner- und der große für die Hunderterstellen zuständig. Die Tausenderstelle wird durch einen Punkt dargestellt. Das Variometer zeigt die Steiggeschwindigkeit in 100 ft/Minute.

Um Ihr Flugzeug zu steuern, gibt es zwei Wege: mit dem Joystick in Port 2 oder über die Tastatur. In Bild 3 sehen Sie die Bedienung mit dem Joystick. Alle Funktionen sind allerdings nicht mit dem Joystick aufrufbar. Die Blickrichtung aus dem Cockpit läßt sich nur über die Tastatur ändern. Die Belegung sehen Sie in Bild 4. Die Bedeutung der restlichen Tasten, mit denen Sie das Flugzeug steuern können, finden Sie in der Tabelle. Auch während der Steuerung mit dem Joystick ist es möglich, die Tastatur zu benutzen.

Um Fliegen zu lernen, ist es am besten, wenn Sie es einfach ausprobieren. Bei eventuellen Abstürzen – des Flugzeugs und nicht des Computers – wird Ihnen nichts passieren, außer einem Farbfimmern des Bildschirms. Am besten ist die sanfte Methode: Also nicht zu heftig gegensteuern, wenn Sie die Flugrichtung ändern wollen, sonst kommt das Flugzeug schnell ins Trudeln.

Wenn Sie den Bogen heraus haben, können Sie noch viele Parameter für Ihren Flug verändern. Dazu drücken Sie während der Simulation <Esc>. In dem CLI-Fenster, von dem aus Sie Flusi gestartet haben, erscheint ein Menü. Damit Sie es ganz sehen, vergrößern Sie das Fenster vor dem Start auf volle Bildschirmgröße.

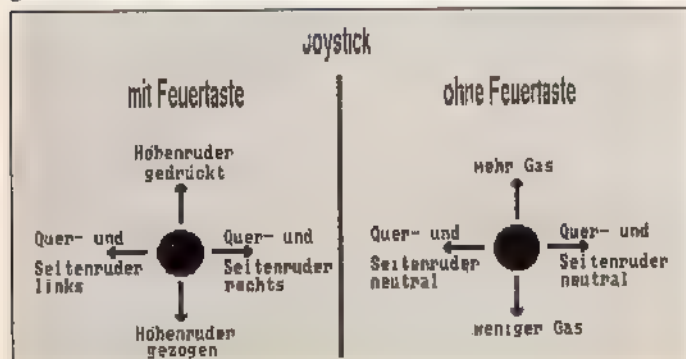


Bild 3. Die Steuerung von »Flusi« mit dem Joystick


1 bis 8

Mit diesen Menüpunkten können Sie verschiedene Grafikelemente ein- oder ausschalten. Das geschieht durch Eingabe der Menüennung und nachfolgendem <RETURN>. Zur nochmaligen Umschaltung wiederholen Sie die Eingabe.

K

Normalerweise sind Quer- und Seitenruder miteinander gekoppelt. Wenn Sie also eines davon bewegen, wird das andere automatisch verstellt. Ist die Kopplung ausgeschaltet, steuern Sie mit dem Joystick nur das Querruder. Das Seitenruder ist dann nur noch über die Tastatur zu beeinflussen.

4	6	Querruder
2	8	Höhenruder
3	9	Gashebe
0		Seitenruder
5		Seiten- und Querruder neutral
1		Bremse
Esc		Menü



Die Tasten, mit denen Sie »Flusi« steuern können

X, Y, Z

Diese drei Parameter bestimmen Ihre Position. Die Grenzwerte stehen hinter der Bezeichnung in Klammern. Geben Sie Werte außerhalb dieser Grenzen ein, verlangt Flusi die erneute Eingabe.

S

Der Steigungswinkel ist der Winkel, in dem Ihr Flugzeug im Verhältnis zur Erde fliegt. 90 Grad entspricht einem Flug senkrecht nach oben. Bei -90 Grad fliegen Sie direkt auf den Erdboden zu. Für einen waagerechten Flug geben Sie als Wert 0 ein.

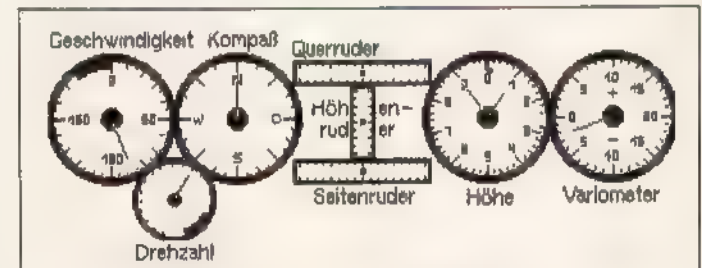


Bild 2. Die Instrumente von »Flusi« mit Erklärung

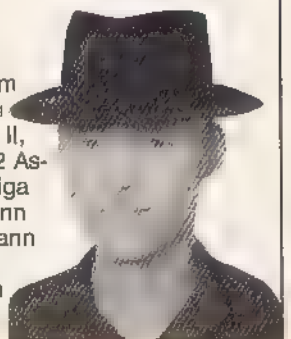
N

Mit diesem Menüpunkt bestimmen Sie die Himmelsrichtung, in die Sie fliegen wollen. Folgende Werte entsprechen den vier Himmelsrichtungen (alle Werte dazwischen sind erlaubt):

0°	Nord
90°	Ost
-180°, 180°	Süd
-90°	West

Michael Koch

Die »Computerkarriere« von Michael Koch begann im Jahr 1983 auf einem Sharp PC-1401 in Basic. Doch dabei blieb es nicht. 1985 folgte ein Apple II, auf dem er nach Basic auch in 6502 Assembler programmierte. Seinen Amiga 1000 kaufte er 1987. Auch hier begann er mit Basic und Assembler, stieg dann aber auf C um. Seine Hobbies sind Astronomie, Elektronik und natürlich der Computer.



ist schöner



H

Der Hängewinkel entspricht der Lage Ihres Flugzeuges. Positive Werte entsprechen einer Schräglage nach rechts, negative einer Lage nach links. Bei 180 und -180 Grad »liegt« Ihr Flugzeug auf dem Rücken.

G, L, B

Diese Werte sind für Fluggeschwindigkeit (G) und Landebahnabmessungen (Länge, Breite) zuständig.

U, I

Dies sind zwei feste Positionen. Nach Auswahl von »U« befindet sich das Flugzeug auf der Startbahn, und Sie können vom Start weg alles selbst erledigen. Zum Üben des Landeanflugs geben Sie »I« ein. Ihr Flugzeug befindet sich dann in der Luft und bewegt sich fast korrekt auf die Landebahn zu. Ihre Aufgabe ist es dann »nur« noch, Ihr Fluggerät heil zu Boden zu bringen.

W, Q

Um mit den veränderten Werten weiterfliegen zu können, wählen Sie nach den Einstellungen im Menü den Punkt »W« an. Sie befinden sich dann wieder im Flugzeug und können weitermachen. Zum Verlassen von Flusi geben Sie <Q> ein. Das Programm endet dann.

Flusi ist multitaskingfähig. Da jedoch die Bildschirme des Programms ständig aktiviert und nach vorne gebracht werden, sind Eingaben in anderen Fenstern unmöglich. Ein Programm, das keine Eingaben benötigt, kann aber im Hintergrund laufen. Das funktioniert gut, da Flusi beim Fliegen nur ungefähr 160 KByte Speicher benötigt.

Ihr Flugzeug trägt auch einen größeren Motor, sprich eine 68020-Karte. Die Geschwindigkeit beim Fliegen steigert sich dann noch einmal. Natürlich ist Flusi keine perfekte Simulation, aber es zeigt deutlich, was mit dem Amiga möglich ist. Die wichtigsten Ei-

genschaften sind berücksichtigt und machen diese Simulation zu einem Programm, das man sicher oft und lange spielen wird. Gut-
René Beaupoil/kn

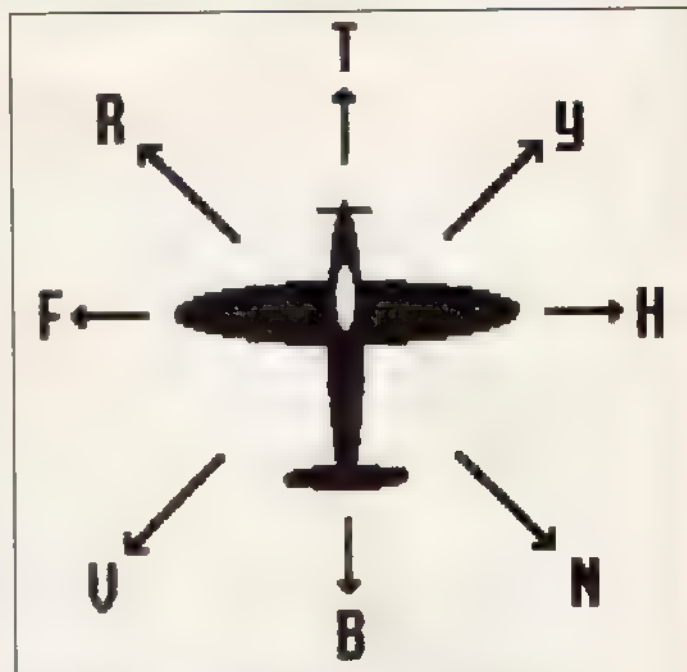


Bild 4. Die verschiedenen Blickrichtungen aus dem Flugzeug und die entsprechenden Tasten

Programmname:	Flusi
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	CC Flusi; LN Flusi.o -LM -LC

Programmautor: Michael Koch

```

1 DCO /* Flugsimulator F L U S I 1.0 Michael Koch März 1989 */
2 5x #include <exec/types.h>
3 3X #include <exec/libraries.h>
4 5h #include <intuition/intuition.h>
5 MA #include <intuition/intuitionbase.h>
6 PQ #include <stdio.h>
7 1j #define CLEAR printf("%c",12)
8 n0 #define PF printf
9 a5 struct IntuitionBase *IntuitionBase,
10 w2 struct GfxBase *GfxBase;
11 QC struct Library *OpenLibrary();
12 nH struct Screen *screen2,*screen3,*screen4,*OpenScreen();
13 Zh struct Window *window1,*window2,*window3,*window4,*OpenWin-
    dow();
14 x8 struct RastPort *rp,*rp2,*rp3,*rp4;
15 xR struct IntuiMessage *nachricht,*GetMsg();
16 5m struct NewScreen newscreen =
17 111 { 0,0,320,160,3,0,1,NULL,CUSTOMSCREEN,NULL,NULL,NULL
    };
18 n0 struct NewWindow newwindow =
19 041 { 0,0,320,160,0,1,VANILLAKEY,BORDERLESS | NOCAREREFRESH,
20 2b3 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CUSTOMSCRE-
    EN };
21 b0 struct NewScreen newscreen4 =
22 P6L { 0,163,320,96,2,0,1,NULL,CUSTOMSCREEN,NULL,NULL,NULL,
    L };
23 JU0 struct NewWindow newwindow4 =
24 041 { 0,0,320,96,0,1,NULL,BORDERLESS | NOCAREREFRESH,
25 pg3 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CUSTOMSCRE-
    EN };
26 nH struct AreaInfo a_info;
27 Zf struct TmpRas t_ras;
28 xX USHORT code;

```

```

29 Ka CWORD areabuf[250],*AllocMem(),*nlaus;
30 w1 BYTE *area_plane,*AllocRaster();
31 7M char input[40];
32 NF float ph,p1,zp,t01,t02,t03,t04;
33 xd float st,hs,az,cst,sbl,sha,cha,obl,ast,saz,cas;
34 6A float pit,nenner,hea,ban,siban,coban,sibha,cohes,sipit,cop
    it;
35 1T float abs1,abco,winkel,weg,geschw,altgeschw,vario,
36 rX float sin(),cos(),tan(),atan(),sqrt();
37 VC int knueppel,taste,c,fenster,det = 0xffff,ende = 0;
38 2H int t1,t2,t3,t4,t5,t6,t7,t8,t9,dx1,dy1,dz1,dx2,dy2,dz2;
39 6R int projx1,projx2,projy1,projy2,punkt1,punkt2,ptk1,ptk2;
40 4u int hruder,qruder,srunder,upa,gas,flag,flag1,flag2;
41 C4 int bahnx1,bahnx2,bahnx3,bahnx4,bahny1,bahny2,bahny3,bah
    ny4;
42 9D int bahnx5,bahnx6,bahnx7,bahnx8,bahny5,bahny6,bahny7,bah
    ny8;
43 tw long transx1,transy1,transx2,transy2,transz2;
44 vH long tx1,tx2,ty1,ty2,tx1,tx2,txalt2,txalt2,txalt2,k;
45 .6 long flugx,flugy,flugs,altflugs,pot,abst,abst2,count = 0;
46 n0 long x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,x21,x22,x23,x24,x25,x26,x27,x28,x29,x30,x31,x32,x33,x34,x35,x36,x37,x38,x39,x40,x41,x42,x43,x44,x45,x46,x47,x48,x49,x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x60,x61,x62,x63,x64,x65,x66,x67,x68,x69,x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x80,x81,x82,x83,x84,x85,x86,x87,x88,x89,x90,x91,x92,x93,x94,x95,x96,x97,x98,x99,x100,x101,x102,x103,x104,x105,x106,x107,x108,x109,x110,x111,x112,x113,x114,x115,x116,x117,x118,x119,x120,x121,x122,x123,x124,x125,x126,x127,x128,x129,x130,x131,x132,x133,x134,x135,x136,x137,x138,x139,x140,x141,x142,x143,x144,x145,x146,x147,x148,x149,x150,x151,x152,x153,x154,x155,x156,x157,x158,x159,x160,x161,x162,x163,x164,x165,x166,x167,x168,x169,x170,x171,x172,x173,x174,x175,x176,x177,x178,x179,x180,x181,x182,x183,x184,x185,x186,x187,x188,x189,x190,x191,x192,x193,x194,x195,x196,x197,x198,x199,x200,x201,x202,x203,x204,x205,x206,x207,x208,x209,x210,x211,x212,x213,x214,x215,x216,x217,x218,x219,x220,x221,x222,x223,x224,x225,x226,x227,x228,x229,x230,x231,x232,x233,x234,x235,x236,x237,x238,x239,x240,x241,x242,x243,x244,x245,x246,x247,x248,x249,x250,x251,x252,x253,x254,x255,x256,x257,x258,x259,x260,x261,x262,x263,x264,x265,x266,x267,x268,x269,x270,x271,x272,x273,x274,x275,x276,x277,x278,x279,x280,x281,x282,x283,x284,x285,x286,x287,x288,x289,x290,x291,x292,x293,x294,x295,x296,x297,x298,x299,x300,x301,x302,x303,x304,x305,x306,x307,x308,x309,x310,x311,x312,x313,x314,x315,x316,x317,x318,x319,x320,x321,x322,x323,x324,x325,x326,x327,x328,x329,x330,x331,x332,x333,x334,x335,x336,x337,x338,x339,x340,x341,x342,x343,x344,x345,x346,x347,x348,x349,x350,x351,x352,x353,x354,x355,x356,x357,x358,x359,x360,x361,x362,x363,x364,x365,x366,x367,x368,x369,x370,x371,x372,x373,x374,x375,x376,x377,x378,x379,x380,x381,x382,x383,x384,x385,x386,x387,x388,x389,x390,x391,x392,x393,x394,x395,x396,x397,x398,x399,x400,x401,x402,x403,x404,x405,x406,x407,x408,x409,x410,x411,x412,x413,x414,x415,x416,x417,x418,x419,x420,x421,x422,x423,x424,x425,x426,x427,x428,x429,x430,x431,x432,x433,x434,x435,x436,x437,x438,x439,x440,x441,x442,x443,x444,x445,x446,x447,x448,x449,x450,x451,x452,x453,x454,x455,x456,x457,x458,x459,x460,x461,x462,x463,x464,x465,x466,x467,x468,x469,x470,x471,x472,x473,x474,x475,x476,x477,x478,x479,x480,x481,x482,x483,x484,x485,x486,x487,x488,x489,x490,x491,x492,x493,x494,x495,x496,x497,x498,x499,x500,x501,x502,x503,x504,x505,x506,x507,x508,x509,x510,x511,x512,x513,x514,x515,x516,x517,x518,x519,x520,x521,x522,x523,x524,x525,x526,x527,x528,x529,x530,x531,x532,x533,x534,x535,x536,x537,x538,x539,x540,x541,x542,x543,x544,x545,x546,x547,x548,x549,x550,x551,x552,x553,x554,x555,x556,x557,x558,x559,x560,x561,x562,x563,x564,x565,x566,x567,x568,x569,x570,x571,x572,x573,x574,x575,x576,x577,x578,x579,x580,x581,x582,x583,x584,x585,x586,x587,x588,x589,x590,x591,x592,x593,x594,x595,x596,x597,x598,x599,x600,x601,x602,x603,x604,x605,x606,x607,x608,x609,x610,x611,x612,x613,x614,x615,x616,x617,x618,x619,x620,x621,x622,x623,x624,x625,x626,x627,x628,x629,x630,x631,x632,x633,x634,x635,x636,x637,x638,x639,x640,x641,x642,x643,x644,x645,x646,x647,x648,x649,x650,x651,x652,x653,x654,x655,x656,x657,x658,x659,x660,x661,x662,x663,x664,x665,x666,x667,x668,x669,x670,x671,x672,x673,x674,x675,x676,x677,x678,x679,x680,x681,x682,x683,x684,x685,x686,x687,x688,x689,x690,x691,x692,x693,x694,x695,x696,x697,x698,x699,x700,x701,x702,x703,x704,x705,x706,x707,x708,x709,x710,x711,x712,x713,x714,x715,x716,x717,x718,x719,x720,x721,x722,x723,x724,x725,x726,x727,x728,x729,x730,x731,x732,x733,x734,x735,x736,x737,x738,x739,x740,x741,x742,x743,x744,x745,x746,x747,x748,x749,x750,x751,x752,x753,x754,x755,x756,x757,x758,x759,x760,x761,x762,x763,x764,x765,x766,x767,x768,x769,x770,x771,x772,x773,x774,x775,x776,x777,x778,x779,x780,x781,x782,x783,x784,x785,x786,x787,x788,x789,x790,x791,x792,x793,x794,x795,x796,x797,x798,x799,x800,x801,x802,x803,x804,x805,x806,x807,x808,x809,x810,x811,x812,x813,x814,x815,x816,x817,x818,x819,x820,x821,x822,x823,x824,x825,x826,x827,x828,x829,x830,x831,x832,x833,x834,x835,x836,x837,x838,x839,x840,x841,x842,x843,x844,x845,x846,x847,x848,x849,x850,x851,x852,x853,x854,x855,x856,x857,x858,x859,x860,x861,x862,x863,x864,x865,x866,x867,x868,x869,x870,x871,x872,x873,x874,x875,x876,x877,x878,x879,x880,x881,x882,x883,x884,x885,x886,x887,x888,x889,x890,x891,x892,x893,x894,x895,x896,x897,x898,x899,x900,x901,x902,x903,x904,x905,x906,x907,x908,x909,x910,x911,x912,x913,x914,x915,x916,x917,x918,x919,x920,x921,x922,x923,x924,x925,x926,x927,x928,x929,x930,x931,x932,x933,x934,x935,x936,x937,x938,x939,x940,x941,x942,x943,x944,x945,x946,x947,x948,x949,x950,x951,x952,x953,x954,x955,x956,x957,x958,x959,x960,x961,x962,x963,x964,x965,x966,x967,x968,x969,x970,x971,x972,x973,x974,x975,x976,x977,x978,x979,x980,x981,x982,x983,x984,x985,x986,x987,x988,x989,x990,x991,x992,x993,x994,x995,x996,x997,x998,x999,x1000,x1001,x1002,x1003,x1004,x1005,x1006,x1007,x1008,x1009,x1010,x1011,x1012,x1013,x1014,x1015,x1016,x1017,x1018,x1019,x1020,x1021,x1022,x1023,x1024,x1025,x1026,x1027,x1028,x1029,x1030,x1031,x1032,x1033,x1034,x1035,x1036,x1037,x1038,x1039,x1040,x1041,x1042,x1043,x1044,x1045,x1046,x1047,x1048,x1049,x1050,x1051,x1052,x1053,x1054,x1055,x1056,x1057,x1058,x1059,x1060,x1061,x1062,x1063,x1064,x1065,x1066,x1067,x1068,x1069,x1070,x1071,x1072,x1073,x1074,x1075,x1076,x1077,x1078,x1079,x1080,x1081,x1082,x1083,x1084,x1085,x1086,x1087,x1088,x1089,x1090,x1091,x1092,x1093,x1094,x1095,x1096,x1097,x1098,x1099,x1100,x1101,x1102,x1103,x1104,x1105,x1106,x1107,x1108,x1109,x1110,x1111,x1112,x1113,x1114,x1115,x1116,x1117,x1118,x1119,x1120,x1121,x1122,x1123,x1124,x1125,x1126,x1127,x1128,x1129,x1130,x1131,x1132,x1133,x1134,x1135,x1136,x1137,x1138,x1139,x1140,x1141,x1142,x1143,x1144,x1145,x1146,x1147,x1148,x1149,x1150,x1151,x1152,x1153,x1154,x1155,x1156,x1157,x1158,x1159,x1160,x1161,x1162,x1163,x1164,x1165,x1166,x1167,x1168,x1169,x1170,x1171,x1172,x1173,x1174,x1175,x1176,x1177,x1178,x1179,x1180,x1181,x1182,x1183,x1184,x1185,x1186,x1187,x1188,x1189,x1190,x1191,x1192,x1193,x1194,x1195,x1196,x1197,x1198,x1199,x1200,x1201,x1202,x1203,x1204,x1205,x1206,x1207,x1208,x1209,x1210,x1211,x1212,x1213,x1214,x1215,x1216,x1217,x1218,x1219,x1220,x1221,x1222,x1223,x1224,x1225,x1226,x1227,x1228,x1229,x1230,x1231,x1232,x1233,x1234,x1235,x1236,x1237,x1238,x1239,x1240,x1241,x1242,x1243,x1244,x1245,x1246,x1247,x1248,x1249,x1250,x1251,x1252,x1253,x1254,x1255,x1256,x1257,x1258,x1259,x1260,x1261,x1262,x1263,x1264,x1265,x1266,x1267,x1268,x1269,x1270,x1271,x1272,x1273,x1274,x1275,x1276,x1277,x1278,x1279,x1280,x1281,x1282,x1283,x1284,x1285,x1286,x1287,x1288,x1289,x1290,x1291,x1292,x1293,x1294,x1295,x1296,x1297,x1298,x1299,x1300,x1301,x1302,x1303,x1304,x1305,x1306,x1307,x1308,x1309,x1310,x1311,x1312,x1313,x1314,x1315,x1316,x1317,x1318,x1319,x1320,x1321,x1322,x1323,x1324,x1325,x1326,x1327,x1328,x1329,x1330,x1331,x1332,x1333,x1334,x1335,x1336,x1337,x1338,x1339,x1340,x1341,x1342,x1343,x1344,x1345,x1346,x1347,x1348,x1349,x1350,x1351,x1352,x1353,x1354,x1355,x1356,x1357,x1358,x1359,x1360,x1361,x1362,x1363,x1364,x1365,x1366,x1367,x1368,x1369,x1370,x1371,x1372,x1373,x1374,x1375,x1376,x1377,x1378,x1379,x1380,x1381,x1382,x1383,x1384,x1385,x1386,x1387,x1388,x1389,x1390,x1391,x1392,x1393,x1394,x1395,x1396,x1397,x1398,x1399,x1400,x1401,x1402,x1403,x1404,x1405,x1406,x1407,x1408,x1409,x1410,x1411,x1412,x1413,x1414,x1415,x1416,x1417,x1418,x1419,x1420,x1421,x1422,x1423,x1424,x1425,x1426,x1427,x1428,x1429,x1430,x1431,x1432,x1433,x1434,x1435,x1436,x1437,x1438,x1439,x1440,x1441,x1442,x1443,x1444,x1445,x1446,x1447,x1448,x1449,x1450,x1451,x1452,x1453,x1454,x1455,x1456,x1457,x1458,x1459,x1460,x1461,x1462,x1463,x1464,x1465,x1466,x1467,x1468,x1469,x1470,x1471,x1472,x1473,x1474,x1475,x1476,x1477,x1478,x1479,x1480,x1481,x1482,x1483,x1484,x1485,x1486,x1487,x1488,x1489,x1490,x1491,x1492,x1493,x1494,x1495,x1496,x1497,x1498,x1499,x1500,x1501,x1502,x1503,x1504,x1505,x1506,x1507,x1508,x1509,x1510,x1511,x1512,x1513,x1514,x1515,x1516,x1517,x1518,x1519,x1520,x1521,x1522,x1523,x1524,x1525,x1526,x1527,x1528,x1529,x1530,x1531,x1532,x1533,x1534,x1535,x1536,x1537,x1538,x1539,x1540,x1541,x1542,x1543,x1544,x1545,x1546,x1547,x1548,x1549,x1550,x1551,x1552,x1553,x1554,x1555,x1556,x1557,x1558,x1559,x1560,x1561,x1562,x1563,x1564,x1565,x1566,x1567,x1568,x1569,x1570,x1571,x1572,x1573,x1574,x1575,x1576,x1577,x1578,x1579,x1580,x1581,x1582,x1583,x1584,x1585,x1586,x1587,x1588,x1589,x1590,x1591,x1592,x1593,x1594,x1595,x1596,x1597,x1598,x1599,x1600,x1601,x1602,x1603,x1604,x1605,x1606,x1607,x1608,x1609,x1610,x1611,x1612,x1613,x1614,x1615,x1616,x1617,x1618,x1619,x1620,x1621,x1622,x1623,x1624,x1625,x1626,x1627,x1628,x1629,x1630,x1631,x1632,x1633,x1634,x1635,x1636,x1637,x1638,x1639,x1640,x1641,x1642,x1643,x1644,x1645,x1646,x1647,x1648,x1649,x1650,x1651,x1652,x1653,x1654,x1655,x1656,x1657,x1658,x1659,x1660,x1661,x1662,x1663,x1664,x1665,x1666,x1667,x1668,x1669,x1670,x1671,x1672,x1673,x1674,x1675,x1676,x1677,x1678,x1679,x1680,x1681,x1682,x1683,x1684,x1685,x1686,x1687,x1688,x1689,x1690,x1691,x1692,x1693,x1694,x1695,x1696,x1697,x1698,x1699,x1700,x1701,x1702,x1703,x1704,x1705,x1706,x1707,x1708,x1709,x1710,x1711,x1712,x1713,x1714,x1715,x1716,x1717,x1718,x1719,x1720,x1721,x1722,x1723,x1724,x1725,x1726,x1727,x1728,x1729,x1730,x1731,x1732,x1733,x1734,x1735,x1736,x1737,x1738,x1739,x1740,x1741,x1742,x1743,x1744,x1745,x1746,x1747,x1748,x1749,x1750,x1751,x1752,x1753,x1754,x1755,x1756,x1757,x1758,x1759,x1760,x1761,x1762,x1763,x1764,x1765,x1766,x1767,x1768,x1769,x1770,x1771,x1772,x1773,x1774,x1775,x1776,x1777,x1778,x1779,x1780,x1781,x1782,x1783,x1784,x1785,x1786,x1787,x1788,x1789,x1790,x1791,x1792,x1793,x1794,x1795,x1796,x1797,x1798,x1799,x1800,x1801,x1802,x1803,x1804,x1805,x1806,x1807,x1808,x1809,x1810,x1811,x1812,x1813,x1814,x1815,x1816,x1817,x1818,x1819,x1820,x1821,x1822,x1823,x1824,x1825,x1826,x1827,x1828,x1829,x1830,x1831,x1832,x1833,x1834,x1835,x1836,x1837,x1838,x1839,x1840,x1841,x1842,x1843,x1844,x1845,x1846,x1847,x1848,x1849,x1850,x1851,x1852,x1853,x1854,x1855,x1856,x1857,x1858,x1859,x1860,x1861,x1862,x1863,x1864,x1865,x1866,x1867,x1868,x1869,x1870,x1871,x1872,x1873,x1874,x1875,x1876,x1877,x1878,x1879,x1880,x1881,x1882,x1883,x1884,x1885,x1886,x1887,x1888,x1889,x1890,x1891,x1892,x1893,x1894,x1895,x1896,x1897,x1898,x1899,x1900,x1901,x1902,x1903,x1904,x1905,x1906,x1907,x1908,x1909,x1910,x1911,x1912,x1913,x1914,x1915,x1916,x1917,x1918,x1919,x1920,x1921,x1922,x1923,x1924,x1925,x1926,x1927,x1928,x1929,x1930,x1931,x1932,x1933,x1934,x1935,x1936,x1937,x1938,x1939,x1940,x1941,x1942,x1943,x1944,x1945,x1946,x1947,x1948,x1949,x1950,x1951,x1952,x1953,x1954,x1955,x1956,x1957,x1958,x1959,x1960,x1961,x1962,x1963,x1964,x1965,x1966,x1967,x1968,x1969,x1970,x1971,x1972,x1973,x1974,x1975,x1976,x1977,x1978,x1979,x1980,x1981,x1982,x1983,x1984,x1985,x1986,x1987,x1988,x1989,x1990,x1991,x1992,x1993,x1994,x1995,x1996,x1997,x1998,x1999,x2000,x2001,x2002,x2003,x2004,x2005,x2006,x2007,x2008,x2009,x2010,x2011,x2012,x2013,x2014,x2015,x2016,x2017,x2018,x2019,x2020,x2021,x2022,x2023,x2024,x2025,x2026,x2027,x2028,x2029,x2030,x2031,x2032,x2033,x2034,x2035,x2036,x2037,x2038,x2039,x2040,x2041,x2042,x2043,x2044,x2045,x2046,x2047,x2048,x2049,x2050,x2051,x2052,x2053,x2054,x2055,x2056,x2057,x2058,x2059,x2060,x2061,x2062,x2063,x2064,x2065,x2066,x2067,x2068,x2069,x2070,x2071,x2072,x2073,x2074,x2075,x2076,x2077,x2078,x2079,x2080,x2081,x2082,x2083,x2084,x2085,x2086,x2087,x2088,x2089,x2090,x2091,x2092,x2093,x2094,x2095,x2096,x2097,x2098,x2099,x2100,x2101,x2102,x2103,x2104,x2105,x2106,x2107,x2108,x2109,x2110,x2111,x2112,x2113,x2114,x2115,x2116,x2117,x2118,x2119,x2120,x2121,x2122,x2123,x2124,x2125,x2126,x2127,x2128,x2129,x2130,x2131,x2132,x2133,x2134,x2135,x2136,x2137,x2138,x2139,x2140,x2141,x2142,x2143,x2144,x2145,x2146,x2147,x2148,x2149,x2150,x2151,x2152,x2153,x2154,x2155,x2156,x2157,x2158,x2159,x2160,x2161,x2162,x2163,x2164,x2165,x2166,x2167,x2168,x2169,x2170,x2171,x2172,x2173,x2174,x2175,x2176,x2177,x2178,x2179,x2180,x2181,x2182,x2183,x2184,x2185,x2186,x2187,x2188,x2189,x2190,x2191,x2192,x2193,x2194,x2195,x2196,x2197,x2198,x2199,x2200,x2201,x2202,x2203,x2204,x2205,x2206,x2207,x2208,x2209,x2210,x2211,x2212,x2213,x2214,x2215,x2216,x2217,x2218,x2219,x2220,x2221,x2222,x2223,x2224,x2225,x2226,x2227,x2228,x2229,x2230,x2231,x2232,x2233,x2234,x2235,x2236,x2237,x2238,x2239,x2240,x2241,x2242,x2243,x2244,x2245,x2246,x2247,x2248,x2249,x2
```



```

64 ny      Folgende globale Variablen stehen zur Verfuegung und
           muessen
65 A36      in Flugmechanik() neu berechnet werden :
66 Bw5      long flux,fluy,fluga      Koordinaten des Flugzeug
           s in cm
67 w        float st                  Steigungswinkel (-PI/2..
           .PI/2)
68 d5       float as                  Azimutwinkel (-PI...P
           I)
69 XX       float ha                  Haengewinkel (-PI...P
           I)
70 Pe       int upm                   Motor-Drehzahl in 1/min
           (0..4000)
71 JT       float geschw              Geschwindigkeit in Knoten
           n
72 tP       Es ist wichtig, dass die Winkel sich innerhalb der an
           gegebenen
73 Lu       Bereiche befinden */
74 sR2      st -= 0.000015 * hruder * cha * geschw; /* Normalflug
           und Rueckenflug */
75 T1       st -= 0.000005 * sruder * sha * geschw; /* Steilkurven
           */
76 os       st += (geschw - 90) * 0.0003; /* Geschwindigkeitsabha
           engigkeit */
77 Rx       if (geschw < 50 && flugz > 202) st -= (50 - geschw) *
           0.01; /* Stall */
78 qM       if (st > ph || st < -ph){ /* Looping-Korre
           ktur */
79 eS4       st = pi - st; if (st > ph) st -= pi;
80 IB       as += pi; if (as > pi) as -= pi;
81 o        ha += pi; if (ha > pi) ha -= pi;
82 Ap2
83 LR       ha += 0.0006 * qruder; /* Einfluss des Querrud
           ers */
84 Ah       ha += 0.0003 * sruder; /* Einfluss des Seitenr
           uders */
85 rK       if (ha > pi) ha -= pi;
86 MA       if (ha < -pi) ha += pi;
87 C3       if (ha < 1 && ha > -1) ha *= 0.995; /* --> stabl
           le Fluglage */
88 51       az += 0.000001 * sruder * cha * geschw; /* Einfluss de
           s Seitenruders */
89 XT       az += 0.000001 * hruder * sha * geschw; /* Einfluss de
           s Höhenruders */
90 35       if (flugz < 202) as += 0.0001 * sruder; /* Bugrad-Len
           kung */
91 1o       if (ha < ph && ha > -ph) az += 0.0004 * ha * geschw; /
           * Hsengev.-Einfl. */
92 2g       while (as > pi) as -= pi;
93 QN       while (as < -pi) as += pi;
94 o8       sat = sin(st); cst = cos(st); sas = sin(as); cas = cos(a
           s);
95 hH       sha = sin(ha); cha = cos(ha); /* wichtig fuer weite
           re Berechnungen */
96 vc       upm = 500 + gas * 20; /* Drehza
           hl in 1/min */
97 rf       if (geschw > 100) upm += (geschw - 100) * 20; /* erhoe
           hte Drehzahl */
98 J4       altgeschw = geschw;
99 hK       geschw = 20 + upm * .04 - 200 * sst; /* Geschwindigk
           eit in Knoten */
100 kb      if (flugz <= 200 && gas == 0) geschw = 0; /* ausrollen
           */
101 wT       geschw = altgeschw + (geschw - altgeschw) / 175;
102 TW       veg = seiddiff * geschw * .0000514; /* Weg in cm */
103 vk       flux += veg * cst * sas; /* neue Koordin
           aten in cm */
104 b9       fluy += veg * cst * caz;
105 87       flugz += veg * sst;
106 d4       if (flugz < 202){
107 J24      flugz = 200; /* tiefer geht es nicht !
           */
108 It       if (st < -0.1 || ha > 0.1 || ha < -0.1) crash();
109 1o       ha = 0; st = 0; /* Bodenkontakt */
110 mH2
111 nID
112 92      rudr(,
113 1E
114 eK      #asm
115 n92      move.w $dfr00c, knueppel /* Joystick-Abfrag
           e */
116 dK      move.w $bfe000, taste
117 XRD     #endasm
118 dQ2     if (taste & 0x0080)[
119 P7       if (knueppel & 2) qruder += 8; /* Ruderstellu
           ngen aendern */
120 p8      if (qruder > 128) qruder = 128;
121 mC      if (knueppel & 512) qruder -= 8;
122 HZ      if (qruder < -128) qruder = -128;
123 VE      if (((knueppel >> 1) ^ knueppel) & 1) hruder += 4;
124 Z1      if (hruder < -128) hruder = -128;
125 ze      if (((knueppel >> 1) ^ knueppel) & 256) hruder += 4;
126 KY      if (hruder > 128) hruder = 128;
127 Cv      if (det & 0x0100) sruder = qruder;
128 d22
129 1d
130 t34     else{
131 gw      if (((knueppel >> 1) ^ knueppel) & 1) gas -= 4; /*
           Gas aendern */
132 sE      if (gas < 0) gas = 0;
133 VK      if (((knueppel >> 1) ^ knueppel) & 256) gas += 4;
134 oc      if (gas > 100) gas = 100;
135 Eg2      if (knueppel & 514) qruder = sruder = 0; /* Ruder in N
           eutralstellung */
136 h0
137 rD      landschaft()
138 ad
139 J02     pit = cst * sbl * sha - cbl * sst; /* Pit
           ch */
140 7K      if (pit >= 1.0 || pit <= -1.0)
141 dS4      (pit > 0) ? (pit = -ph) : (pit = ph);
142 sH2     else /* arcsin */
143 914      pit = -atan(pit / sqrt(1.0 - pit * pit));
144 v12     hea = cha * sbl; /* Hea
           ding */
145 7n      nenner = cbl * cst - sbl * sst * sha;
146 su      if (nenner != 0){
147 kh      hea = as + atan(hea / nenner);
148 hF      if (nenner < 0) hea += pi;
149 Pu2     }
150 3y     else{
151 9M4      if (cha > 0)
152 Gb6      (sbl > 0) ? (hea = as + ph) : (hea = as - ph);
153 354     else
154 Mn6      (sbl > 0) ? (hea = as - ph) : (hea = as + ph);
155 V02
156 ag      while (hea > pi) hea -= pi;
157 CN      if (hea < -pi) hea += pi;
158 ns      ban = cbl * sha + sbl * sst; /* Ban
           k */
159 Tx      nenner = cha * cst;
160 A4      if (nenner == 0)
161 2a4      (ban < 0) ? (ban = ph) : (ban = -ph);
162 rA2     else{
163 No4      ban = -atan(ban / nenner);
164 dc      if (nenner < 0)
165 SM      { ban += pi; if (ban > pi) ban -= pi; }
166 gB2     }
167 HS      siban = sin(ban); coban = cos(ban);
168 IW      sihea = sin(hea); cohea = cos(hea);
169 12      sipit = sin(pit); copit = cos(pit);
170 51      t01 = cohea * coban; t02 = -sihea * coban; /* Transfor
           mationsmatrix */
171 3T      t03 = -cohea * siban; t04 = sihea * siban;
172 pw      t1 = (int) (4096 * (t01 - t04 * sipit));
173 gw      t2 = (int) (4096 * (t02 + t03 * sipit));
174 w4      t3 = (int) (4096 * (copit * siban));
175 OJ      t4 = (int) (8192 * (t03 + t02 * sipit));
176 FR      t5 = (int) (8192 * (t04 - t01 * sipit));
177 dB      t6 = (int) (8192 * (copit * coban));
178 sS      t7 = (int) (4096 * (sihea * copit));
179 69      t8 = (int) (4096 * (cohea * copit));
180 Dk      t9 = (int) (4096 * (sipit));
181 JS      flag = flag1 + flag2 = 0; /* Horizont in 5000km Ent
           fernung */
182 ob      x3 = 500000000 * sihea; y3 = 500000000 * cohea;
183 tC      x4 = y3 << 1, y4 = x3 << 1;
184 Md      x1 = x3 x4, y1 = y3 + y4; x2 = x3 + x4; y2 = y3 - y4,
           z1 = z2 = 0;
185 p9      kante(0);
186 sV      if (flag){
187 H14      SetAPen(rp,71),
188 JH      RectFill(rp,0L,0L,319L,159L); /* altes Bild loes

```

Listing. Ein toller Flugsimulator mit allem Drum und Dran: »Flusi«. Bitte mit dem Checksummer (Seite 143) eingeben.


```

chen */
189 Ia SetAPen(rp,1L),
190 Ad projx1 = pktx; projy1 = pkt; ecke(); /* Land ...
191 FN AreaEnd(rp);
192 b62 ]
193 kF else
194 eR4 (pit > 0) ? (SetAPen(rp,7L)) : (SetAPen(rp,1L));
195 LX RectFill(rp,0L,0L,319L,159L); /* alles fuenen */
196 A62 ]
197 Dk if (det & 0x08) { /* Stadt */
198 Hi4 flag = flag1 = flag2 = 0;
199 DJ z1 = z2 = 0;
200 Tk x1 = -200000; y1 = -200000; x2 = -100000; y2 = -600000
, kante(0);
201 MQ x1 = -100000; y1 = -600000; x2 = -600000; y2 = 900000
; kante(1);
202 y6 x1 = -600000; y1 = -900000; x2 = -500000; y2 = -100000
; kante(1);
203 TF x1 = -500000; y1 = -100000; x2 = -200000; y2 = -200000
; kante(1);
204 Am if (flag) {
205 yI6 projx1 = pktx; projy1 = pkt; ecke();
206 mQ SetAPen(rp,2L); AreaEnd(rp);
207 Lq4 ]
208 Mx2 }
209 Ir if ((det & 0x01) && (pit < 0.7)) { /* Erdboden mit Gitter
r versehen */
210 Np4 (sihea > 0) ? (abs1 = sihea) : (abs1 = -sihea);
211 K6 (cohea > 0) ? (abco = cohea) : (abco = -cohea);
212 Im x3 = x5 = flugx + flugs * (0.7 * sipit + copit * (4 *
sihea - 2 * abco));
213 cl x4 = x6 = flugx + flugs * (-0.7 * sipit + copit * (4 *
sihea + 2 * abco));
214 L6 y3 = y5 = flugy + flugs * (0.7 * sipit + copit * (4 *
cohea - 2 * abs1));
215 81 y4 = y6 = flugy + flugs * (-0.7 * sipit + copit * (4 *
cohea + 2 * abs1));
216 FN for (pot = 12500; pot < 100000000; pot <<= 1) {
217 3v6 if (flugs < (pot - (pot >> 2))) {
218 Vq8 abst = pot << 1; abst2 = abst << 3;
219 J7 if (flugs > (pot >> 1))
220 fgA SetAPen(rp,4L);
221 DB8 else
222 V8A SetAPen(rp,0L);
223 bh8 z1 = z2 = 0;
224 YI x3 -= (x3 - pot) % abst; x4 -= (x4 + pot) % abst;
225 lp y3 -= (y3 - pot) % abst; y4 -= (y4 + pot) % abst;
226 6k gitter();
227 eX SetAPen(rp,0L);
228 WJ x3 = x5 - x5 % abst; x4 = x6 - x6 % abst;
229 mF y3 = y5 - y5 % abst; y4 = y6 - y6 % abst;
230 Co gitter();
231 en break;
232 kF6 ]
233 IG4 }
234 mU2 }
235 9U0 #define MAKRO5 {ecke();
236 kQG if (flag)
237 I6I AreaDraw(rp,(long)projx1,(long)projy1);
238 XIG else
239 xBI AreaMove(rp,(long)projx1,(long)projy1);
240 XUG AreaDraw(rp,(long)projx2,(long)projy2);
241 NZ2 if (det & 0x02) { /* Startbahn */
242 zQ6 flag = flag1 = flag2 = 0;
243 v1 z1 = z2 = 0;
244 I6 x1 = -bahnbreite; y1 = -bahnlaenge; x2 = -bahnbreite;
y2 = bahnlaenge;
245 QK if (projektion(0)) {
246 W66 bahnx1=projx1; bahnny1=projy1; bahnx2=projx2; bahnny2=
projy2;
MAKRO5; flag += 1;
247 A6
248 OV4 x2 = bahnbreite; y2 = bahnlaenge;
249 LN if (projektion(1)) {
250 AS bahnx3=projx1; bahnny3=projy1; bahnx4=projx2; bahnny4=
projy2;
251 r66 MAKRO5; flag += 2;
252 *

```

```

253 5a4
254 qi x2 = bahnbreite; y2 = -bahnlaenge;
255 oX if (projektion(1)) {
256 CA6 bahnx5=projx1; bahnny5=projy1; bahnx6=projx2; bahnny6=
projx2;
MAKRO5; flag += 1;
257 TS
258 Af2
259 2Y x2 = -bahnbreite; y2 = -bahnlaenge;
260 au if (projektion(1)) {
261 Xd6 bahnx7=projx1; bahnny7=projy1; bahnx8=projx2; bahnny8=
projx2;
MAKRO5; flag += 8;
262 xm
263 Fk4
264 SK
265 wH6
266 I1 if (flag)
projx1 = pktx; projy1 = pkt; ecke();
SetAPen(rp,2L); /* graue Startba
hn */
AreaEnd(rp);
267 tk
268 Ep4
269 VD SetAPen(rp,5L); /* Weiße Linien */
270 VZ if (flag & 1) {
271 X16 Move(rp,(long)bahnx1,(long)bahnny1); Draw(rp,(long)ba
hnx2,(long)bahnny2);
272 Qt4
273 L if (flag & 2) {
274 vH6 Move(rp,(long)bahnx3,(long)bahnny3); Draw(rp,(long)ba
hnx4,(long)bahnny4);
275 Ru4
276 G9 if (flag & 4) {
277 X16 Move(rp,(long)bahnx5,(long)bahnny5); Draw(rp,(long)ba
hnx6,(long)bahnny6);
278 Uz4
279 VS if (flag & 8) {
280 7F6 Move(rp,(long)bahnx7,(long)bahnny7); Draw(rp,(long)ba
hnx8,(long)bahnny8);
281 X24
282 PA if (flugs < 100000 && flugs > -30000 && flugs < 300
0) {
283 b66 x1 = x2 = 0;
/* Mittellinie */
284 z if (flugs > 20000) { /* Flughoehe ueber 200m */
285 OT8 y1=bahnlaenge; y2=bahnlaenge; linie();
286 e76
287 NV else /* Flughoehe unter 200m */
288 Tu8 y3 = flugy - (flugs << 2) - 5000;
289 yE y4 = flugy + (flugs << 2) + 15000;
290 vY y3 -= y3 % 10000; y4 -= y4 % 10000;
291 Q7 if ((y3 <= -bahnlaenge) || (y3 >= bahnlaenge))
292 VaA y1=-bahnlaenge; y2=bahnlaenge; linie();
293 jE8
294 NI e.se
295 CuA 1: (y3 <= -bahnlaenge) y3 = -bahnlaenge;
296 PK else
297 wDC y1 = -bahnlaenge; /* durchgezogene Linie
am neg. Ende */
y2 = y3 - 3000; linie();
298 AQ
299 pXA
300 7y if (y4 >= bahnlaenge) y4 = bahnlaenge;
301 UP else
302 kwC y1 = bahnlaenge; /* durchgezogene Linie
am pos. Ende */
y2 = y4; linie();
303 gx
304 uFA
305 r if (y1 < y3 && y1 < y4 && y1 <= 10000)
306 OTC y2 = y1+7000; linie(); /* die Striche a
ind 70m lang */
307 XLA
308 J78
309 zU6
310 OV4
311 1a2
312 FT if (det & 0x04) { /* See */
313 6Z4 flag = flag1 = flag2 = 0;
314 4A z1 = z2 = 0;
315 G6 x1 = 200000; y1 = 300000; x2 = 500000; y2 = 600000; ka
nte(0);
x1 = 500000; y1 = 600000; x2 = 800000; y2 = 200000; ka
nte(1);
x1 = 800000; y1 = 200000; x2 = 400000; y2 = -200000; ka
nte(1);
x1 = 400000; y1 = -200000; x2 = 200000; y2 = 300000; ka
nte(1);
316 S1 if (flag)
317 GU projx1 = pktx; projy1 = pkt; ecke();
318 mQ
319 Id
320 uA6

```



```

321 pJ      SetAPen(rp,6L); AreaEnd(rp);
322 Ch4
323 Ad      ,
324 Ek      flag = flag1 = flag2 = 0;          /* Insel */
325 ao      z1 = z2 = 0;
326          x1 = 350000; y1 = 250000; x2 = 500000; y2 = 250000; ka
nte(0);
327 gl      x1 = 500000; y1 = 250000; x2 = 450000; y2 = 100000; ka
nte(1);
328 Au      x1 = 450000; y1 = 100000; x2 = 350000; y2 = 250000; ka
nte(1);
329 HR6      if (flag){
330 f8        projx1 = pktx,projy1 = pkt;ecke();
331 lq4        SetAPen(rp,1L); AreaEnd(rp);
332 M-2
333 9a      if (det & 0x20){
334 VC4        x3 = 350000; y3 = -300000; windrad(,
335 f8        x3 = 400000; y3 = -300000; windrad(,
336 nZ        x3 = 450000; y3 = -250000; windrad(,
337 4        x3 = -200000; y3 = 200000; windrad(,
338 bx2
339 Bt      if (det & 0x10){
340 l54        x1 = -200000; y1 = 500000; funkturm();
341 FR        x1 = -350000; y1 = 250000; funkturm();
342 J        x1 = -400000; y1 = 100000; funkturm();
343 KC        x1 = -350000; y1 = -250000; funkturm();
344 bI        x1 = -250000; y1 = -550000; funkturm();
345 KU        x1 = -450000; y1 = -650000; funkturm();
346 Ad        x1 = 650000; y1 = -500000; funkturm();
347 Jx        x1 = 450000; y1 = 200000; funkturm();
348 c72
349 aT      if (det & 0x40){          /* Wald mit 10 Baeumen */
350 2W4        SetAPen(rp,0L);
351 T8        x1 = 204000; y1 = 112000; baum(); x1 = 210000; y1 = 11
2000; baum();
352 y        x1 = 190000; y1 = 110000; baum(); x1 = 198000; y1 = 10
8000; baum();
353 ym        x1 = 206000; y1 = 104000; baum(); x1 = 196000; y1 = 10
2000; baum();
354 L        x1 = 200000; y1 = 100000; baum(); x1 = 192000; y1 = 9
6000; baum();
355 oJ        x1 = 204000; y1 = 94000; baum(); x1 = 196000; y1 = 9
0000; baum();
356 KF2
357 DL      if (det & 0x80){          /* Vogel */
358 rO4      if (vogel){
359 PE6        if ((vogel++) == 35) vogel = 0; /* Der Vogel ist
weg */
360 JM        else{
361 xh8          SetAPen(rp,0L);
362 dv          x1 = vogelx; x2 = vogelx + 50; y1 = y2 = vogely;
363 Nq          s1 = vogels; (vogelx2) ? (s2 = s1 + 20) : (s2 = s1
- 20);
364 tK          linie(); x2 = vogelx - 50; linie();
365 xH          vogely -= 300; /* Eigenbewegung des Vo
gels */
366 JF6
367 vQ4
368 ZU
369 B6      else
370 9TB      if ((flugs & 511L) == 511L && flugs < 100000){ /*
neuer Vogel ? */
371 xL        vogel = 1;
372 6X        vogelx = flugs + s2 + 20000 + (flugs & 8191);
373 Og        vogely = flugs + s2 + 20000 + (flugs & 8191);
374 v6        vogels = flugs + 500 + (flugs & 1023);
375 JY6        if (vogels < 200) vogels = 200;
376 4Z4
377 a2
378 JH      x1 = x2 = 0; s1 = s2 = 500000000; y1 = y2 = -s2; /* S
onne im Sueden */
379 HJ      SetAPen(rp,5L);
380 Gh      if (projektion(0))
381 ye4      RectFill(rp,(long)projx1-1,(long)projy1-2,(long)projx1
+1,(long)projy1);
382 8n      /****** Platz fuer Landschafts-Erweiterungen *****/
383 Bg0
384 9J      kante(modus)
385 q0      int modus;
386 Ad
387 lI2      if (projektion(modus)){
388 v54        ecke();
389 sC        if (flag AreaDraw(rp,(long)projx1,(long)projy1),

```

```

390 HK      else AreaMove(rp,(long)projx1,(long)projy1);
391 Jq      AreaDraw(rp,(long)projx2,(long)projy2);
392 O7      flag = 1;
393 lq2
394 v-0
395 Jb      ecke() /* eventuelles Auffuellen der Ecken beim Flaech
henfuellen */
396 An
397 ks2      if (flag2){ /* Quellpunkt ist schon vorhanden */
398 lI4        if (punkt == 0){
399 B36          if (projy1 == 0) AreaDraw(rp,0L,0L);
400 L-        if (projx1 == 319)
401 aq          [ AreaDraw(rp,0L,0L);AreaDraw(rp,319L,0L); ]
402 4C          if (projy1 == 159)
403 FO          [ AreaDraw(rp,0L,0L);AreaDraw(rp,319L,0L);AreaDraw(r
p,319L,159L); ]
404 vI4
405 JA
406 vJ6      if (punkt == 0){
407 vH          if (projx1 == 319) AreaDraw(rp,319L,0L);
408 vK          if (projy1 == 159)
409 p0          [ AreaDraw(rp,319L,0L);AreaDraw(rp,319L,159L); ]
410 vT          if (projx1 == 0)
411 d84          [ AreaDraw(rp,319L,0L);AreaDraw(rp,319L,159L);AreaDr
aw(rp,0L,159L); ]
412 DO
413 kY6      if (punkt == 319){
414 uL          if (projy1 == 159) AreaDraw(rp,319L,159L);
415 BF          if (projx1 == 0)
416 lT          [ AreaDraw(rp,319L,159L);AreaDraw(rp,0L,159L); ]
417 RD          if (projy1 == 0)
418 KP4          [ AreaDraw(rp,319L,159L);AreaDraw(rp,0L,159L);AreaDr
aw(rp,0L,0L); ]
419 MH
420 v-6      if (punkt == 159){
421 oI          if (projx1 == 0) AreaDraw(rp,0L,159L);
422 l-          if (projy1 == 0)
423 vZ          [ AreaDraw(rp,0L,159L);AreaDraw(rp,0L,0L); ]
424 FO          if (projx1 == 319)
425 vH4          [ AreaDraw(rp,0L,159L);AreaDraw(rp,0L,0L);AreaDraw(r
p,319L,0L); ]
426 vF
427 vO2      flag2 = 0; /* Jetzt ist kein Quellpunkt mehr da */
428 nd
429 lB4      if ((projx2==0) || (projx2==319) || (projy2==0) || (proj
y2==159)){
430 lY        flag2 = 1; /* Jetzt ist wieder ein Quellpunkt da */
431 x82        punktx = projx2; punkty = projy2;
432 CG
433 a84      if (flag1 == 0){ /* falls kein Zielpunkt da ist */
434 cL6          if ((projx1==0) || (projx1==319) || (projy1==0) || (pr
ojy1==159)){
435 9L          flag1 = 1; /* Jetzt ist ein Zielpunkt da */
436 2X4          pktx = projx1; pkt; = projy1;
437 JY2
438 4Z0
439 OZ      gitter()
440 vV
441 D2      if (x3 > flugs) x3 == abst;
442 8      if (x4 < flugs) x4 += abst;
443 xH      if (y3 > flugs) y3 == abst;
444 xH      if (y4 < flugs) y4 += abst;
445 vI          x1 = x3 - abst2; x2 = x4 + abst2;
446 L-          for (y1 = y3; y1 <= y4; y1 += abst)
447 vI5          [ y2 = y1; linie(); ]
448 qI2          y1 = y3 - abst2; y2 = y4 + abst2;
449 JK          for (x1 = x3; x1 <= x4; x1 += abst)
450 v-5          [ x2 = x1; linie(); ]
451 Hm0
452 H8      baum()
453 F1
454 vZ2      x1 = 0; s2 = 2000; x2 = x1; y2 = y1; linie();
455 og      if (flugs > x1-40000 && flugs < x1+40000 && flugs > y
1-40000 \
456 9b6          && flugs < y1+40000 && flugs < 40000){
457 TH4          x1 += 500; s2 -= 500; x2 = x1 + 500; y2 = y1 + 300; l1
nie();
458 OY          x1 += 300;          x2 = x1;          y2 = y1 - 500; l1
nie();
459 Zy          s1 += 300;          x2 = x1 - 400; y2 = y1 + 200; l1
nie();
460 Qv2

```

Listing. Ein toller Flugsimulator: »Flusi« (Fortsetzung)


```

461 Rn0
462 EG Funktion(,
463 Ts
464 eg2 SetAPen(rp,5L);
465 Fq z1 = 0; z2 = 14000; x2 = x1, y2 = y1; linie();
466 cq if (flux > x1-100000 && flux < x1+100000 && flugy >
y1-100000 \
&& flugy < y1+100000 && flux < 100000){
467 IW6 SetAPen(rp,3L);
468 ob4 for (z1=1500; z1<11000; z1+=3000) /* rote Streife
469 QM n */
z2 = z1 + 1500; linie();
470 Lt7
471 b62
472 o70
473 F7 windrad(,
474 a3
475 pr2 SetAPen(rp,5L),
476 gu switch (count % 3){
477 2K4 case 0 : x1 = x3 + 2000; x2 = x3 - 2000, /* Flue
gel */
y1 = y3 + 2000; y2 = y3 - 2000,
478 OmD z1 = 3000; z2 = 7000; linie(); break;
479 F5 case 1 : x1 = x3 + 2732; x2 = x3 - 2732;
y1 = y3 + 2732; y2 = y3 - 2732;
480 Y24 z1 = 5732; z2 = 4268; linie(); break;
481 IdD case 2 : x1 = x3 + 732; x2 = x3 - 732;
y1 = y3 + 732; y2 = y3 - 732,
482 Iv z1 = 7732; z2 = 2268; linie(); break;
483 JD4
484 oeD }
485 Jw
486 qL2 }
487 FZ if (flux > x1-300000 && flux < x1+300000 && flugy >
y1-300000 \
&& flugy < y1+300000 && flux < 300000){
488 x56 SetAPen(rp,3L); /* T
489 yU4 urn */
z1 = 0; x2 = x3 + 500; y2 = y3 - 500; z2 = 5000,
490 Hr x1 = x3 + 1000; y1 = y3; linie();
491 IF y1 = y3 - 1000; linie();
492 DO x1 = x3; y1 = y3 - 1000; linie();
493 II x1 = x3; y1 = y3; linie();
494 7L z1 = 5000; x2 = x3 + 1000; y2 = y3 - 1000; linie();
495 R
496 OV2
497 L40
498 Jy linie()
499 2S
500 XR2 if (projektion(0)){
501 LS4 Move(rp,(long)projx1,(long)projy1);
502 to Draw(rp,(long)projx2,(long)projy2);
503 7c2
504 840 }
505 So projektion.modus
506 x int modus,
507 7a
508 do2 if (modus == 1)
509 s1 [ tx1 = txalt2; ty1 = tyalt2; ts1 = tsalt2; ]
510 rw else{
511 6e4 deltax1 = x1 - flux; deltay1 = y1 - flugy; deltas1 =
z1 - flugz,
512 Ah0 #define MAKRO1 [ deltax1 >= 2; deltay1 >= 2; deltas
1 >= 2; ]
while (deltax1 > 32767 || deltax1 < -32768) MAKRO1;
513 eB4 while (deltay1 > 32767 || deltay1 < -32768) MAKRO1;
514 OI while (deltas1 > 32767 || deltas1 < -32768) MAKRO1;
515 6V dx1 = (int)deltax1; dy1 = (int)deltay1; dz1 = (int)del
ta1,
516 zI transx1 = dx1 * (long)t1 + dy1 * (long)t2 + dz1 * (lon
g)t3,
517 wJ transy1 = dx1 * (long)t4 + dy1 * (long)t5 + dz1 * (lon
g)t6,
518 w3 transz1 = dx1 * (long)t7 + dy1 * (long)t8 + dz1 * (lon
g)t9;
519 6n
520 S70 #define MAKRO2 [ transx1 >= 2; transy1 >= 2; transz
1 >= 2; ]
while (transx1 > 32767 || transx1 < -32768) MAKRO2;
521 w64 while (transy1 > 32767 || transy1 < -32768) MAKRO2;
522 eC while (transz1 > 32767 || transz1 < -32768) MAKRO2;
523 mQ tx1 = (int)transx1; ty1 = (int)transy1; ts1 = (int)tra
nsz1;
524 qP
525 Ty2 deltax2 = x2 - flux; deltay2 = y2 - flugy; deltas2 = z2
- flugz;
526 Fe
527 oQ0 #define MAKRO3 [ deltax2 >= 2; deltay2 >= 2; deltas
2 >= 2; ]
while (deltax2 > 32767 || deltax2 < -32768) MAKRO3;
528 Sif2 while (deltay2 > 32767 || deltay2 < -32768) MAKRO3;
529 ap

```

```

530 fx while (deltas2 > 32767 || deltas2 < -32768) MAKRO3;
531 eI dx2 = (int)deltax2; dy2 = (int)deltay2; dz2 = (int)delta
z2,
532 Rn transx2 = dx2 * (long)t1 + dy2 * (long)t2 + dz2 * (long)
t3,
533 IX transy2 = dx2 * (long)t4 + dy2 * (long)t5 + dz2 * (long)
t6,
534 bH transz2 = dx2 * (long)t7 + dy2 * (long)t8 + dz2 * (long)
t9,
535 6q0 #define MAKRO4 [ transx2 >= 2; transy2 >= 2; transz
2 >= 2; ]
while (transx2 > 32767 || transx2 < -32768) MAKRO4;
536 6a2 while (transy2 > 32767 || transy2 < -32768) MAKRO4;
537 EK while (transz2 > 32767 || transz2 < -32768) MAKRO4;
538 Mh tx2 = (int)transx2; ty2 = (int)transy2; tz2 = (int)trans
z2;
539 IY txalt2 = tx2; tyalt2 = ty2; tzalt2 = tz2;
for (j;j)
540 LB c = 0;
541 yF
542 C94 if (tx1 < -ts1) c+= 1; if (tx1 > ts1) c+= 2;
543 p if (ty1 < -ts1) c+= 4; if (ty1 > ts1) c+= 8;
544 M7 if (tx2 < -tz2) c+= 16; if (tx2 > tz2) c+= 32;
545 J if (ty2 < -tz2) c+= 64; if (ty2 > tz2) c+= 128;
546 rn
547 Lq if (c == 0){
548 ZV6 if (ts1 == 0) ts1 = 1; /* 3D --
> 2D */
if (ts2 == 0) ts2 = 1;
549 xD projx1 = (319 + (tx1 * 319) / ts1) >> 1;
550 Cr projy1 = (159 - (ty1 * 159) / ts1) >> 1;
551 OA projx2 = (319 + (tx2 * 319) / ts2) >> 1;
552 I3 projy2 = (159 - (ty2 * 159) / ts2) >> 1;
553 EM return(1);
554 TP break;
555 s1
556 yT4
557 Io if ((c&17)==17) || ((c&34)==34) || ((c&68)==68) || ((
c&136)==136)){
return(0); break; /* Linie ist nicht sichtbar */
558 g66
559 1M4
560 Nk if (c & 1),
k = ((ts1+tx1) << 15) / (tx1-tx2-tz2+ts1);
561 d06 tx1 = (k * (ts1-ts2) >> 15) - ts1;
562 zI ty1 = (k * (ty2-ty1) >> 15) + ty1,
563 15 ts1 = -tx1; continue;
564 ee
565 7c4
566 Ya
567 qV6 if (c & 2)
k = ((ts1-tx1) << 15) / (tx2-tx1-ts2+ts1);
568 wL tx1 = (k * (ts2-ts1) >> 15) + ts1;
569 oB ty1 = (k * (ty2-ty1) >> 15) + ty1,
570 FK ts1 = tx1; continue;
571 D14
572 oE
573 1M6 if (c & 4){
k = ((ts1+ty1) << 15) / (ty1-ty2-ts2+ts1);
574 ex tx1 = (k * (tx2-tx1) >> 15) + tx1;
575 rm ty1 = (k * (ts1-ts2) >> 15) - ts1;
576 w ts1 = -ty1; continue;
577 oC4
578 h;
579 wV6 if (c & 8){
k = ((ts1-ty1) << 15) / (ty2-ty1-ts2+ts1);
580 kJ tx1 = (k * (tx2-tx1) >> 15) + tx1;
581 Co ty1 = (k * (ts2-ts1) >> 15) + ts1;
582 nb ts1 = ty1; continue;
583 Pu4
584 mS
585 9J6 if (c & 16){
k = ((ts2+tx2) << 15) / (tx2-tx1-ts1+ts2);
586 w2 tx2 = (k * (ts2-ts1) >> 15) - ts2,
587 Dh ty2 = (k * (ty1-ty2) >> 15) + ty2;
588 C9 ts2 = -tx2; continue;
589 V04
590 eJ
591 Kx6 if (c & 32),
k = ((ts2-tx2) << 15) / (tx1-tx2-ts1+ts2);
592 hX tx2 = (k * (ts1-ts2) >> 15) + ts2;
593 In ty2 = (k * (ty1-ty2) >> 15) + ty2;
594 y; ts2 = tx2; continue,
595 b64
596 B4
597 KA6 if (c & 64){
k = ((ts2+ty2) << 15) / (ty2-ty1-ts1+ts2);
598 9Z tx2 = (k * (tx1-tx2) >> 15) + tx2;
599 mG ty2 = (k * (ts2-ts1) >> 15) - ts2;
600 TR ts2 = -ty2; continue;
601 wC4
602 40
603 1M6 if (c & 128){
k = ((ts2-ty2) << 15) / (ty1-ty2-ts1+ts2);
604 FR tx2 = (k * (tx1-tx2) >> 15) + tx2;
605 nE ty2 = (k * (ts1-ts2) >> 15) + ts2;

```



```

606 EA      tz2 = ty2; continue;
607 nI4    }
608 oJ2    ]
609 pK0    }
610 kF      tastatur()
611 nG
612 EX2    while (nachricht = GetMag(window3->UserPort)){
613 3x4      ReplyMag(nachricht);
614 QJ      if (nachricht->Class == VANILLAKEY){
615 xD6      code = nachricht->Code;
616 6w      if (code > 95) code -= 32; /* Umwandlung in grosse
Buchstaben */
617 9C      switch (code){
618 x08      case '1' : if (flux < 202) geschw -= 2; /* Br
eschw = /
if (geschw < 0) geschw = 0; break;
case '4' : qruder -= 8; if (qruder < -128) qruder =
-128; break;
case '6' : qruder += 8; if (qruder > 128) qruder =
128; break;
case '2' : hruder -= 8; if (hruder < -128) hruder =
-128; break;
case '8' : hruder += 8; if (hruder > 128) hruder =
128; break;
case '5' : qruder = sruder = 0; break;
case '3' : gas -= 4; if (gas < 0) gas = 0; break;
case '9' : gas += 4; if (gas > 100) gas = 100; bre
ak;
case '0' : sruder -= 8; if (sruder < -128) sruder =
-128; break;
case '.' : sruder += 8; if (sruder > 128) sruder =
128; break;
case 'T' : sbl = 0.000; cbl = 1.000; break; /* Bl
ickrichtungen */
case 'Y' : sbl = 0.707; cbl = 0.707; break;
case 'H' : sbl = 1.000; cbl = 0.000; break;
case 'N' : sbl = 0.707; cbl = -0.707; break;
case 'S' : sbl = 0.000; cbl = -1.000; break;
case 'V' : sbl = -0.707; cbl = -0.707; break;
case 'F' : sbl = -1.000; cbl = 0.000; break;
case 'R' : sbl = 0.707; cbl = 0.707; break;
case 27 : menu(); break;
} /* Ende von switch */
if (det & 0x0100) sruder = qruder,
640 Kp4    }
641 Lq2    ]
642 Nr0    ]
643 6h      long eing(min,max)
644 zh      long min,max;
645 Lo      [
646 wa2      long neuwert;
647 Ka      do{
648 Jk4      PF(" Neuer Wert (%ld...%ld) : ",min,max);
649 Nx      gets(input); neuwert = atol(input);
650 bT2      } while (neuert < min || neuwert > max);
651 NZ      return(neuert);
652 W10
653 oQ      menu()
654 Ux      {
655 HE2      WBenchToFront();
656 D7      ActivateWindow(window1);
657 F10      anzeigen:
658 dF2      CLEAR;
659 9y      PF(" "); for (abst = 0; abst < 75; abst++) PF(" ");
660 Ze      PF("\n          F L U S I 1. 0\n");
661 80      PF(" "); for (abst = 0; abst < 75; abst++) PF(" ");
662 ux      PF("\n\n");
663 np      PF(" ( 1 ) Gitternetz. . . . . an/aus : %d\n", (de
t & 0x01) == 0x01);
664 4q      PF(" ( 2 ) Startbahn. . . . . an/aus : %d\n", (de
t & 0x02) == 0x02);
665 7H      PF(" ( 3 ) See+Insel. . . . . an/aus : %d\n", (de
t & 0x04) == 0x04);
666 Fk      PF(" ( 4 ) Stadt. . . . . an/aus : %d\n", (de
t & 0x08) == 0x08);
667 En      PF(" ( 5 ) Punktuerne. . . . . an/aus : %d\n", (de
t & 0x10) == 0x10);
668 x4      PF(" ( 6 ) Windraeder. . . . . an/aus : %d\n", (de
t & 0x20) == 0x20);
669 EW      PF(" ( 7 ) Wald. . . . . an/aus : %d\n", (de
t & 0x40) == 0x40);
670 14      PF(" ( 8 ) Voegel. . . . . an/aus : %d\n", (de
t & 0x80) == 0x80);
671 JB      PF(" ( K ) Kopplang Seiren+Queeruder an/aus : %d\n", (de

```

```

t & 0x100) == 0x100);
672 Zg      PF(" ( X ) X-Koordinate in Meter (-100000...100000): %l
d\n", flux/100);
673 kC      PF(" ( Y ) Y-Koordinate in Meter (-100000...100000): %l
d\n", fluxy/100);
674 pq      PF(" ( Z ) Flughoehe in Meter ( 2...100000) : %l
d\n", fluxz/100);
675 56      PF(" ( S ) Steigungswinkel in Grad (-90.....90): %d
\n", (int)(st*57.3));
676 98      PF(" ( A ) Azimutwinkel in Grad (-180.....180): %d
\n", (int)(as*57.3));
677 w6      PF(" ( H ) Haengewinkel in Grad (-180.....180, %d
\n", (int)(hs*57.3));
678 Wx      PF(" ( G ) Geschwindigkeit in Knoten ( 0 .. 200): %d
\n", (int)geschw);
679 Tk      PF(" ( L ) Landebahn-Laenge in Meter (1000..100000): %l
d\n", bahnlange/50);
680 7H      PF(" ( B ) Landebahn-Breite in Meter ( 10.....200): %l
d\n", bahnbreite/50);
681 St      PF(" ( U ) Startposition\n",
682 2e      PF(" ( I ) Landeanflug\n",
683 qJ      PF(" ( W ) Weiterfliegen\n",
684 Kk      PF(" ( Q ) Programm beenden\n");
685 yy      PF(" Ihre Eingabe : ");
686 J4      gets(input); PF("\n");
687 OP      if (input[0] > 95) input[0] -= 32; /* Wandlung in gros
se Buchstaben */
688 vq      switch (input[0]){
689 S14      case '1' : det = 0x0001; break;
690 bu      case '2' : det = 0x0002; break;
691 pB      case '3' : det = 0x0004; break;
692 De      case '4' : det = 0x0008; break;
693 b2      case '5' : det = 0x0010; break;
694 pC      case '6' : det = 0x0020; break;
695 iR      case '7' : det = 0x0040; break;
696 Lq      case '8' : det = 0x0080; break;
697 oV      case 'K' : det = 0x0100; break;
698 z5      case 'X' : flux = 100 * eing(-100000L,100000L); break
;
699 5D      case 'Y' : fluxy = 100 * eing(-100000L,100000L); break
;
700 rK      case 'Z' : fluxz = 100 * eing(2L,100000L); break;
701 8s      case 'S' : st = 0.01745 * eing(- 90L, 90L); break;
702 pb      case 'A' : as = 0.01745 * eing(-180L,180L); break;
703 uV      case 'H' : hs = 0.01745 * eing(-180L,180L); break;
704 ds      case 'G' : geschw = eing(0L,200L); break;
705 Ol      case 'L' : bahnlange = 50 * eing(1000L,100000L); brea
k;
706 TZ      case 'B' : bahnbreite = 50 * eing(10L,200L); break;
707 qd      case 'U' : flux = 0; fluxy = -bahnlange; fluxz = 200
;
st = 0; as = 0; hs = 0; sbl = 0; cbl = 1; g
as = 0;
geschw = 0; hruder = qruder = sruder = 0; b
reak;
708 NCF      case 'I' : flux = -10000; fluxy = -1000000; fluxz = 8
5000;
st = -0.09; as = 0.01; hs = -0.01; sbl = 0;
cbl = 1;
gas = 50; geschw = 80; hruder = qruder = sr
uder = 0; break;
709 QP      case 'Q' : ende = 1; CLEAR; PF("\n bye-bye f\n\n"); br
eak;
710 JH4      }
711 MpF      if (input[0] != 'W' && input[0] != 'Q') goto anzeigen;
712 bJ      CurrentTime(&lt;sek,&lt;mikrosek);
713 ha4      WBenchToBack();
714 W12      ActivateWindow(window3);
715 gR      }
716 OC      crash()
717 pL      {
718 JF      SetAPen(rp2,3L); RectFill(rp2,0L,0L,319L,159L);
719 b60      SetAPen(rp3,5L); RectFill(rp3,0L,0L,319L,159L);
720 I1      for (abst = 1; abst < 50; abst++){
721 E2      ScreenToFront(screen2);
722 Ow2      ScreenToFront(screen3);
723 GG      }
724 4N      hruder = sruder = qruder = 0; gas = 0; geschw = 0,
sbl = 0; cbl = 1;
725 7p4      initinstrumente()
726 At      }
727 JE2      }
728 4w      }
729 LW      }
730 mH0      }
731 G6      }
732 kD      }

```

Listing. Ein toller Flugsimulator: »Flusi« (Fortsetzung)


```

733 xg2 SetAPen(rp4,0L);
734 fx RectFill(rp4,0L,0L,319L,95L); /* Hintergrund */
735 jn SetAPen(rp4,1L);
736 ev for (abst = 30; abst < 34; abst++)
737   for (abst2 = 32; abst2 < 300; abst2 += 64)
738     if (abst2 != 160)
739       DrawEllipse(rp4,abst2,34L,abst,abst);
740 z72 for (abst = 19; abst < 22; abst++)
741   DrawEllipse(rp4,64L,73L,abst,abst);
742 l42 SetAPen(rp4,3L);
743 bx for (abst2 = 32; abst2 < 300; abst2 += 64)
744   if (abst2 != 160)
745     AreaEllipse(rp4,abst2,34L,5L,5L);
746 k32 AreaEllipse(rp4,64L,73L,3L,3L);
747 x2 AreaAnd(rp4,
748   ziff(31,12,0), ziff(48,32,5), ziff(52,32,0),
749   ziff(27,52,1), ziff(31,52,0), ziff(35,52,0),
750   ziff(10,32,1), ziff(14,32,5), ziff(18,32,0),
751   zeich(94,32,0); zeich(114,32,1); zeich(94,52,2); zeich
752   (74,32,3);
753 .. ziff(223,12,0); ziff(235,15,1), ziff(23,24,2), ziff(243
754   ,39,3);
755 n0 ziff(235,48,4), ziff(223,52,5), ziff(211,48,6), ziff(203
756   ,39,7);
757 .. ziff(203,24,8), ziff(211,15,9),
758   ziff(264,32,0), ziff(272,18,5), ziff(272,45,5),
759   ziff(284,12,1), ziff(268,12,0), ziff(284,52,1); ziff(288
760   ,52,0);
761 Xb ziff(297,18,1), ziff(301,18,5), ziff(297,45,1), ziff(301
762   ,45,5);
763 pg ziff(304,32,2), ziff(308,32,0),
764 .. Move(rp4,288L,24L); Draw(rp4,288L,20L);
765 lp Move(rp4,286L,22L); Draw(rp4,290L,22L);
766 gM Move(rp4,286L,46L); Draw(rp4,290L,46L);
767 Lm SetAPen(rp4,2L); /* rote kritische Ber
768   eiche */
769 57 for (winkel = 5.60; winkel < 6.283; winkel += .01){
770 Sy4 Move(rp4,(long)(32+sin(winkel)*26),(long)(34-cos(winke
771   l)*26));
772 .. Draw(rp4,(long)(32+sin(winkel)*29),(long)(34-cos(winke
773   l)*29));
774 Mr2 for (winkel = 2; winkel < 2.8; winkel += .01){
775 r1 Move(rp4,(long)(64+sin(winkel)*17),(long)(75-cos(winke
776   l)*17));
777 Aq4 Draw(rp4,(long)(64+sin(winkel)*19),(long)(75-cos(winke
778   l)*19));
779 Qv2 SetAPen(rp4,3L);
780 lX marken(32,34,6,4); marken(32,34,3,20); marken(32,34,
781   1,40);
782 J marken(96,34,6,8); marken(96,34,2,16); marken(96,34,
783   1,32);
784 2n marken(224,34,3,10); marken(224,34,2,50);
785 GA marken(288,34,5,8); marken(288,34,2,40);
786 .. for (winkel = -2.8; winkel < 2.8; winkel += .3)
787 2Q4 WritePixel(rp4,(long)(65+sin(winkel)*18),(long)(76-cos
788   (winkel)*18));
789 vL2 SetAPen(rp4,1L);
790 oJ RectFill(rp4,125L,9L,193L,17L);
791 gF RectFill(rp4,154L,17L,166L,56L);
792 rF RectFill(rp4,125L,59L,193L,67L);
793 rI SetArea(rp4,1L);
794 m1 RectFill(rp4,127L,7L,193L,15L);
795 .. RectFill(rp4,156L,19L,164L,53L);
796 jw RectFill(rp4,127L,57L,193L,65L);
797 Om SetAPen(rp4,3L);
798 xK for (abst = 128; abst < 193; abst += 4)
799   [ WritePixel(rp4,abst,7L); WritePixel(rp4,abst,15L);
800   WritePixel(rp4,abst,57L); WritePixel(rp4,abst,65L); ]
801 lg2 for (abst = 128; abst < 193; abst += 16)
802   [ WritePixel(rp4,abst,8L); WritePixel(rp4,abst,14L);
803   WritePixel(rp4,abst,58L); WritePixel(rp4,abst,64L); ]
804 jH2 for (abst = 20; abst < 53; abst += 4)
805   [ WritePixel(rp4,156L,abst); WritePixel(rp4,164L,abst);
806   for (abst = 20; abst < 53; abst += 8)
807     [ WritePixel(rp4,157L,abst); WritePixel(rp4,163L,abst);
808     SetAPen(rp4,3L);
809     RectFill(rp4,200L,80L,290L,100L);
810     SetAPen(rp4,2L);
811     Move(rp4,210L,90L);
812     SetDrMd(rp4,JAM1);

```

```

802 Nw Text(rp4,"Flusi 1.0\0",9L);
803 x50 }
804 J2 siff(x,y,n)
805 Qn int x,y,n;
806 .. F :
807 vY2 int zeile,spalte,hilf;
808 JU hilf = ziffer[n];
809 w6 for (zeile = 0; zeile < 5; zeile++)
810   k64 for (spalte = 0; spalte < 3; spalte++)
811     o16 if ((hilf <= 1) & 0x8000)
812       Yes WritePixel(rp4,(long) x + spalte,(long) y + zeile)
813       ;
814 7e0 |
815 21 zeich(x,y,n)
816 a2 int x,y,n;
817 o2 ..
818 Re2 int zeile,spalte;
819 F6 long hilf;
820 In hilf = zeichen[n];
821 7H for (zeile = 0; zeile < 5; zeile++)
822   1P4 for (spalte = 0; spalte < 3; spalte++)
823     B16 if ((hilf <= 1) & 0x80000000)
824       jp8 WritePixel(rp4,(long) x + spalte,(long) y + zeile)
825       ;
826 In0 |
827 NK marken(x,y,laenge,anzahl)
828 Bn int x,y,laenge,anzahl,
829 Hn
830 2E2 float winkel,siv,cow;
831 yV laenge = 30 - laenge;
832 B1 for (winkel = 0; winkel < 2pi; winkel += (2pi/anzahl)){
833   1C4 siv = sin(winkel); cow = cos(winkel);
834   mV Move(rp4,(long)(x + siv * laenge),(long)(y - cow * la
835     nge));
836   TE Draw(rp4,(long)(x + siv * 29),(long)(y - cow * 29)
837   );
838   Ex2
839   Ty0 {
840   DO instrumente()
841   Ru {
842   vV2 Move rp4,zeiger0x,zeiger0y; Draw rp4,64L,75L); /*
843     Drehzahlmesser */
844   E5 winkel = upw * .0016 - 2.8;
845   w0 if (winkel > 2.8) winkel = 2.8;
846   m zeiger0x = 64 + (int)(sin(winkel) * 15);
847   WF zeiger0y = 75 - (int)(cos(winkel) * 15);
848   YF Move(rp4,zeiger0x,zeiger0y); Draw(rp4,64L,75L);
849   hL Move(rp4,zeiger1x,zeiger1y); Draw(rp4,224L,34L);
850   /* Hosenmesser */
851   GC Draw(rp4,zeiger2x,zeiger2y);
852   XF winkel = flugs * .0001915;
853   Rn zeiger1x = 224 + (int)(sin(winkel) * 23);
854   Ee zeiger1y = 34 - (int)(cos(winkel) * 23);
855   e7 winkel /= 10;
856   d6 zeiger2x = 224 + (int)(sin(winkel) * 15);
857   Q6 zeiger2y = 34 - (int)(cos(winkel) * 15);
858   ta Move(rp4,zeiger1x,zeiger1y); Draw(rp4,224L,34L);
859   OK Draw(rp4,zeiger2x,zeiger2y);
860   J4 RectFill(rp4,zeiger1x,zeiger1y,zeiger1x+2,zeiger1y+2;
861   ID winkel /= 10;
862   YF zeiger1x = 224 + (int)(sin(winkel) * 23);
863   YF zeiger1y = 33 - (int)(cos(winkel) * 23);
864   Bn RectFill(rp4,zeiger1x,zeiger1y,zeiger1x+2,zeiger1y+2;
865   Pg Move(rp4,zeiger1x,zeiger1y); Draw(rp4,32L,34L); /* G
866     eschwindigkeit */
867   DY winkel = geschw * .031416;
868   q5 if (winkel > 6) winkel = 6; if (winkel < 0) winkel = 0
869   ;
870   wv zeiger1x = 32 + (int)(sin(winkel) * 22);
871   XD zeiger1y = 34 - (int)(cos(winkel) * 22);
872   d8 Move(rp4,zeiger1x,zeiger1y); Draw(rp4,32L,34L);
873   fQ Move(rp4,zeiger5x,zeiger5y); Draw(rp4,288L,34L);
874   /* Variometer */
875   nk winkel = vario * .1571;
876   Fu if (winkel > 3) winkel = 3; if (winkel < -3) winkel = -3
877   ;
878   7E zeiger5x = 288 - (int)(cos(winkel) * 22);
879   EU zeiger5y = 34 - (int)(sin(winkel) * 22);
880   Fm Move(rp4,zeiger5x,zeiger5y); Draw(rp4,288L,34L);
881   gt Move(rp4,zeiger6x,zeiger6y); Draw(rp4,96L,34L);
882   /* Kompass */
883   Oy zeiger6x = 96 + (int)(saz * 22);
884   Jw zeiger6y = 34 - (int)(cas * 22);
885   vA Move(rp4,zeiger6x,zeiger6y); Draw(rp4,96L,34L);

```



```

875 k1  PortFill(rp4,zeiger7x,10L,zeiger7x+2,12L
      /* Querruder */
876 y0  zeiger7x = 159 + (qruder >> 2);
877 92  RectFill(rp4,zeiger7x,10L,zeiger7x+2,12L);
878 n4  RectFill(rp4,1*9,zeiger8y,16L,zeiger8y+2,
      /* Hoehenruder */
879 Dm  zeiger8y = 35 + (hruder >> 3);
880 qe  RectFill(rp4,159L,zeiger8y,16L,zeiger8y+2);
881 sD  RectFill(rp4,zeiger9x,60L,zeiger9x+2,62L);
      /* Seitenruder */
882 KQ  zeiger9x = 159 + (sruder >> 2);
883 Dh  RectFill(rp4,zeiger9x,60L,zeiger9x+2,62L);
884 G10
885 x2  farban(zeiger)
886 JP  BYTE *zeiger;
887 Fi
888 ZP2  SetRGB4(zeiger,0L, 0L, 0L, 0L); /* schwarz starke Lin
ien,Rand */
889 Cp  SetRGB4(zeiger,1L, 0L,12L, 0L); /* gruen Land
      */
890 YL  SetRGB4(zeiger,2L, 9L, 9L, 9L); /* grau Landebahn
      */
891 WK  SetRGB4(zeiger,3L,15L, 0L, 0L); /* rot Windraeder
,Funkturm */
892 Ag  SetRGB4(zeiger,4L, 0L,10L, 0L); /* gruen schwache L
inien */
893 Vy  SetRGB4(zeiger,5L,15L,15L,15L); /* weiss Windreeder
,Tuerme */
894 Na  SetRGB4(zeiger,6L, 0L, 0L,10L); /* blau Wasser
      */
895 cO  SetRGB4(zeiger,7L, 0L, 9L,15L); /* hellblau Himmel
      */
896 Sx0
897 AW  main()
898 Qt
899 is2  GfxBase = (struct GfxBase*) OpenLibrary("graphics.librar
y",0L,
900 Km  if (GfxBase == NULL) goto cleanup;
901 IW  IntuitionBase = (struct IntuitionBase*) OpenLibrary("int
uition.library",0L,
902 Jv  if (IntuitionBase == NULL) goto cleanup;
903 o8  window1 = IntuitionBase->ActiveWindow;
904 Eb  screen2 = OpenScreen(&newscreen);
905 jK  if (screen2 == NULL) goto cleanup;
906 r2  newwindow.Screen = screen2;
907 xp  window2 = OpenWindow(&newwindow);
908 m1  if (window2 == NULL) goto cleanup;
909 SN  rp2 = window2->RPort;
910 nH  Farben(&screen2->ViewPort);
911 Om  screen3 = OpenScreen(&newscreen);
912 sU  if (screen3 == NULL) goto cleanup;
913 OC  newwindow.Screen = screen3;
914 7D  window3 = OpenWindow(&newwindow);
915 vs  if (window3 == NULL) goto cleanup;
916 ha  rp3 = window3->RPort;
917 OQ  farben(&screen3->ViewPort);
918 6p  screen4 = OpenScreen(&newscreen4);
919 1e  if (screen4 == NULL) goto cleanup;
920 nG  newwindow4.Screen = screen4;
921 p3  window4 = OpenWindow(&newwindow4);
922 42  if (window4 == NULL) goto cleanup;
923 vv  rp4 = window4->RPort;
924 Wd  SetRGB4(&screen4->ViewPort,0L, 7L, 7L, 7L); /* greu
Hintergrund */
925 oF  SetRGB4(&screen4->ViewPort,1L, 0L, 0L, 0L); /* schwarz
Instrumente */
926 2s  SetRGB4(&screen4->ViewPort,2L,15L, 0L, 0L); /* rot
kritische Bereiche */
927 JQ  SetRGB4(&screen4->ViewPort,3L,15L,13L, 0L); /* g
Zeiger,Ziffern */
928 1Z  InitArea(&a_info,&arabuf[0],100L,
929 in  area_plane = AllocRaster(320L,256L);
930 dE  if (area_plane == NULL) goto cleanup;
931 n5  rp2->AreaInfo = &a_info; rp2->TapRes = &t_res;
932 wG  rp3->AreaInfo = &a_info; rp3->TapRes = &t_res;
933 5F  rp4->AreaInfo = &a_info; rp4->TapRes = &t_res;
934 Mf  InitTapRes(&t_res,area_plane,(long)RASSIZE(320,256));
935 sY  maus = AllocMem(5L,0x00010003); /* Chip-Ram */
936 Cd  SetPointer(window1,maus,0L,0L,0L,0L); /* Mauszeiger ->g
      */
937 xO  SetPointer(window2,maus,0L,0L,0L,0L);
938 wU  SetPointer(window3,maus,0L,0L,0L,0L);
939 1a  SetPointer(window4,maus,0L,0L,0L,0L);
940 DE  phi=2*atan(1.0); /* PI/2 */

```

```

941 F5  pi=4*atan(1.0); /* PI */
942 cN  zpi=2*atan(1.0); /* 2*PI */
943 4F  InitInstrumente();
944 Xn  SetDirMJ(rp4,COMPLEMENTL.AMI);
945 St  CurrentTime(&altsek,&altmikrosek);
946 rW  startsituation;
947 o2  Flugx = 0; /* Koordinaten des Flugzeugs in cm
      */
948 1W  Flugy = 0;
949 hi  Flugz = 100;
950 pF  s = 0; /* Steigungswinkel (-PI/2.
      */
951 X9  sa = 0; /* Haengewinkel (-PI
      */
952 EC  az = 0; /* Flugrichtungswinkel (-PI
      */
953 AX  sin = 0, cos = 1; /* sin() bzw. cos(Kopfwinkel des Pi
oter */
954 Ex  canllaenge = 500000; /* halbe Laenge der Startbahn in cm
      */
955 xA  ambreite = 2500; /* halbe Breite der Startbahn in cm
      */
956 kx  hruder = 0; /* Stellung des Hoehenruders (-128.
...128 */
957 zf  qruder = 0; /* Stellung des Querruders (-128.
...128 */
958 o9  sruder = 0; /* Stellung des Seitenruders (-128.
...128 */
959 TX  gas = 60; /* Gas in Prozent ( 0.
...100 */
960 St  geschw = 85; /* Geschwindigkeit in Knoten ( 0.
...200 */
961 1i  for(;;){ /* Hauptschleife */
962 Vy4  if (ende) goto cleanup2;
963 JP  cleanup;
964 dx  ruder();
965 4w  ActivateWindow(window3); tastatur();
966 An  CurrentTime(&sek,&mikrosek);
967 yo  zeitdiff = 1000000 * (sek - altsek) + (mikrosek - altm
ikrosek);
968 v2  altsek = sek; altmikrosek = mikrosek;
969 yA  CleanupScreen;
970 17  if (zeitdiff != 0) /* Steiggeschw. in 10
cm/min */
971 Dn6  vario = (19685 * (Flugz - altflugz)) / (float)zeitdi
ff;
972 ts4  altflugz = Flugz;
973 P6  if (fenster == 2){
974 1j6  fenster=3; rp=rp3;
975 78  landschaft(); instrumente(); /* neues Bild verdeck
t zeichnen */
976 Ow  WBenchToBack();
977 Ph  ScreenToBack(screen2); /* und dann darstelle
t */
978 mH4
979 QL  else{
980 xF6  fenster=2; rp=rp2;
981 su  landschaft(); instrumente();
982 62  WBenchToBack();
983 5I  ScreenToBack(screen3);
984 sM4
985 tQ2
986 Hn0  cleanup;
987 12  goto "weiter";
988 th0  cleanup2:
989 Hp2  if (area_plane) FreeRaster(area_plane,320L,256L);
990 ZS  ClearPointer(window1); ClearPointer(window2);
991 uh  ClearPointer(window3); ClearPointer(window4);
992 eg  FreeMem(maus,8L);
993 44  if (window2) CloseWindow (window2);
994 4W  if (screen2) CloseScreen (screen2);
995 CE  if (window3) CloseWindow (window3);
996 CW  if (screen3) CloseScreen (screen3);
997 KO  if (window4) CloseWindow (window4);
998 Kg  if (screen4) CloseScreen (screen4);
999 3d  if (IntuitionBase) CloseLibrary (IntuitionBase);
1000 Xi  if (GfxBase) CloseLibrary (GfxBase);
1001 sy  exit(0);
1002 x70
(C) 1989 M&T

```

Listing. Ein toller Flugsimulator mit allen Schikanen:
»Flusi« (Schluß)

Ping- pong pur

Ping Pong« (Listing 1) füllt eine Lücke im Spieleangebot für den Amiga. Es ist nämlich unseres Wissens nach die erste Tischtennis-Simulation für diesen Computer. Die Programmierung in Assembler verleiht dem Programm eine rasante Geschwindigkeit und bringt Schwung ins Spiel. Die dreidimensionale Grafik und der »klassische« Pingpong-Sound tragen ihren Teil dazu bei, daß das Spiel absolut realitätsnah ist und Sie an den Joystick fesseln wird (Bild 1).

»Ping Pong« ist außerdem eines jener wenigen Spiele, bei denen zwei Spieler gleichzeitig gegeneinander antreten können.

Action zu zweit

Alles, was Sie tun müssen, um den kleinen weißen Ball auch auf Ihrem Bildschirm zum Leben zu erwecken, ist, das Listing mit dem Checksummer einzugeben. Da das Programm in Assembler geschrieben wurde, wir aber möglichst vielen Lesern ermöglichen wollen, in den Genuß von »Ping Pong« zu kommen, haben wir

uns entschlossen, das Listing als Basic-Lader abzdrukken. Es kann also unter Beachtung der Eingabehinweise (siehe Kasten) wie jedes andere Basic-Programm eingetippt werden. Nach dem ersten Start von »PingPong_Gen« wird das lauffähige Programm »Ping Pong« automatisch auf Diskette erzeugt (siehe Eingabehinweise im Kasten).

Das Spiel selbst läuft nach den üblichen Tischtennisregeln ab. Sieger ist der Spieler, der zwei Sätze gewinnt. Ein Satz ist gewonnen, wenn man 21 Punkte erreicht und mindestens zwei Punkte mehr auf seinem Konto als der Gegner hat.

Das Match beginnt mit der Angabe des Spielers im Vordergrund. Wenn er den Knopf an seinem Joystick drückt, fliegt der »simulierte Zelluloidball« in die Höhe, um sich dann gleich wieder dem Tischtennis-Schläger und dem Boden zu nähern. Der Joystick-Knopf muß nun ein weiteres Mal betätigt werden, um den Ball in die gegnerische Hälfte zu befördern — wichtig ist dabei der richtige Augenblick. Nach einigem Experimentie-

Bild 1. Sport, Spiel, Spannung mit der dreidimensionalen Tisch-

Programmname:	PingPong_gen
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic
Bemerkung:	erzeugt lauffähiges Programm auf eingelegerter Diskette (s. Kasten Seite 51)

Programmautor: S. Ören

```

1 0m0 REM Generiert lauffähiges Programm
2 Kq REM >>PingPong<< auf der Diskette
3 eU REM im Laufwerk df0:
4 e1 CLS
5 oJ OPEN "df0:pingpong" FOR OUTPUT AS 1
6 DU READ ans
7 qd FOR i=1 TO ans
8 5p1 READ h$
9 0D2 wert1=ASC(LEFT$(h$,1))
10 dR IF wert1>64 THEN wert1=wert1-87
    ELSE wert1=wert1-48
11 Hk wert1=wert1*16
12 9e wert2=ASC(RIGHT$(h$,1))
13 yr IF wert2>64 THEN wert2=wert2-87
    ELSE wert2=wert2-48
14 Rk wert=wert1+wert2
15 BI PRINT #1,CHR$(wert))
16 lQO NEXT
17 5p CLOSE 1
18 2x END
19 0e Werte:
20 GE DATA 9496
21 rj DATA 00,00,03,f3,00,00,00,00,00,00
22 Se DATA 00,01,00,00,00,00,00,00,00,00
23 aB DATA 00,00,07,e6,00,00,03,e9,00,00
24 h2 DATA 07,e6,2c,78,00,04,20,3c,00,00
25 eV DATA 78,00,22,3c,00,03,00,02,4e,ae

```

```

26 gS DATA ff,3a,67,00,12,f6,23,e0,00,00
27 r0 DATA 1f,30,20,3c,00,00,2e,00,22,3c
28 Bn DATA 00,03,00,02,4e,ae,ff,3a,67,00
29 6m DATA 12,de,23,e0,00,00,1f,3c,2a,7e
30 or DATA 00,00,1a,04,20,3c,00,00,18,c8
31 eY DATA 22,3c,00,00,19,10,2c,3c,00,00
32 q3 DATA 19,58,26,3c,00,00,19,64,28,3c
33 gM DATA 00,00,19,fc,2c,3c,00,00,19,70
34 6x DATA 2e,3c,00,00,19,b8,3d,40,00,06
35 00 DATA 48,40,3d,40,00,02,3d,41,00,0e
36 xk DATA 48,41,3d,41,00,0a,3d,42,00,16
37 lV DATA 48,42,3d,42,00,12,3d,43,00,1e
38 TE DATA 48,43,3d,43,00,1a,3d,44,00,26
39 6z DATA 3d,44,00,2e,48,44,3d,44,00,22
40 TJ DATA 3d,44,00,2a,3d,46,00,36,48,46
41 Q6 DATA 3d,46,00,32,3d,47,00,3a,48,47
42 53 DATA 3d,47,00,3a,20,3a,1e,78,22,00
43 J1 DATA 24,01,06,81,00,00,28,00,06,82
44 Nf DATA 00,00,50,00,23,c1,00,00,1f,34
45 1F DATA 23,e2,00,00,1f,38,26,3a,1e,64
46 Dm DATA 48,40,3d,40,00,42,48,40,3d,40
47 hM DATA 00,46,48,41,3d,41,00,4a,48,41
48 73 DATA 3d,41,00,4e,48,42,3d,42,00,52
49 Bp DATA 48,42,3d,42,00,56,48,43,3d,43
50 l3 DATA 00,5a,48,43,3d,43,00,5e,3c,3c
51 Qr DATA 00,40,06,80,00,00,00,06,80,40
52 4x DATA 20,3c,00,00,00,04,22,3c,00,00
53 xV DATA 00,09,61,00,16,1e,91,fc,00,00
54 vX DATA 00,02,20,3c,00,00,00,04,22,3c
55 RX DATA 00,00,00,0f,61,00,16,08,22,48
56 HT DATA e2,5e,20,3c,00,00,00,06,61,00
57 oD DATA 16,2e,20,49,e2,5e,d1,fc,00,00
58 4m DATA 50,00,20,3c,00,00,00,05,61,00
59 hE DATA 16,1a,22,48,d1,fc,00,00,00,2a
60 xH DATA 20,3c,00,00,00,14,61,00,16,08
61 9Y DATA ee,5e,20,49,d1,fc,00,00,00,2c
62 GE DATA 20,3c,00,00,00,14,61,00,15,f4
63 lQ DATA 20,49,91,fc,00,00,27,d4,20,3c
64 dD DATA 00,00,00,14,3c,3c,22,00,61,00
65 wo DATA 15,de,20,7a,1d,94,d1,fc,00,00
66 4F DATA 06,fa,3c,3c,08,00,20,3c,00,00
67 Tb DATA 00,04,22,3c,00,00,00,0b,61,00

```

```

68 e1 DATA 15,e6,d1,fc,00,00,00,02,20,3c
69 lQ DATA 00,00,00,04,22,3c,00,00,00,0d
70 9d DATA 61,00,15,90,22,48,e3,5e,20,3c
71 HU DATA 00,00,00,06,61,00,15,9c,20,49
72 ZL DATA d1,fc,00,00,50,00,e3,5e,20,3c
73 dJ DATA 00,00,00,05,61,00,15,88,3a,06
74 Vm DATA eb,5e,20,3c,00,00,00,14,22,48
75 vy DATA d1,fc,00,00,00,24,61,00,15,72
76 0G DATA 20,49,91,fc,00,00,27,d4,20,3c
77 6g DATA 00,00,00,14,3c,3c,01,10,61,00
78 MA DATA 15,5e,3c,05,20,49,d1,fc,00,00
79 yF DATA 00,26,20,3c,00,00,00,14,61,00
80 3r DATA 15,48,20,7a,1d,02,3c,3c,00,10
81 VB DATA d1,fc,00,00,06,e8,20,3c,00,00
82 lq DATA 00,04,22,3c,00,00,00,0b,61,00
83 Vb DATA 14,26,91,fc,00,00,00,02,20,3c
84 xD DATA 00,00,00,04,22,3c,00,00,00,0d
85 hX DATA 61,00,14,e0,91,fc,00,00,00,28
86 6w DATA 30,ba,00,00,91,fc,00,00,00,28
87 St DATA 30,ba,00,00,20,7a,1c,ba,d1,fc
88 9K DATA 00,00,06,fa,3c,3c,20,00,20,3c
89 eF DATA 00,00,00,04,22,3c,00,00,00,0d
90 0e DATA 61,00,14,e0,d1,fc,00,00,00,02
91 ha DATA 20,3c,00,00,00,04,22,3c,00,00
92 VL DATA 00,0b,61,00,14,26,91,fc,00,00
93 8a DATA 00,28,30,ba,00,00,91,fc,00,00
94 6M DATA 00,28,30,ba,00,00,20,7a,1c,72
95 R3 DATA d1,fc,00,00,07,18,3c,3c,00,22
96 YQ DATA 20,3c,00,00,00,7a,61,00,14,e0
97 2C DATA 20,7a,1c,5a,31,7a,00,00,06,e8
98 yJ DATA 31,7c,00,00,06,fa,20,7a,1c,46
99 ao DATA 20,3c,00,00,77,fa,42,83,61,00
100 j5 DATA 15,9e,22,3c,00,00,00,0b,20,7a
101 8F DATA 1c,30,d1,fc,00,00,0e,de,3c,3c
102 Ex DATA ff,ff,20,3c,00,00,00,00,61,00
103 qe DATA 14,70,91,fc,00,00,00,26,51,c9
104 20 DATA ff,ea,22,3c,00,00,00,0b,20,7a
105 tF DATA 1c,0c,d1,fc,00,00,0e,de,3c,3c
106 Mu DATA 00,00,20,3c,00,00,00,00,61,00
107 Rb DATA 14,48,91,fc,00,00,00,26,51,c9
108 64 DATA ff,ea,22,3c,00,00,00,0b,20,7a
109 jg DATA 1b,ec,d1,fc,00,00,0f,06,3c,3c

```


Erleben Sie Tischtennis hautnah auf dem Bildschirm. Tückische Angaben, Schmettern und Anschneiden des Balls sind in dieser Simulation realistisch umgesetzt. Bei zwei gleichwertigen Gegnern ergibt das packende Ballwechsel.

ren findet man schnell den günstigsten Zeitpunkt heraus.

Schaffen Sie es, den Ball über das Netz zu befördern, ist der Gegner an der Reihe. Dieser muß ebenfalls den Ball mit seinem Schläger treffen und dabei gleichzeitig den Feuerknopf drücken. Mit der Bewegung des Joysticks während des Schlages nach rechts oder links kann der Abprallwinkel des Balls manipuliert werden. So wird das Anschneiden des Balls realisiert. Wird der Joystick nach vorne bewegt, steigert sich die Geschwindigkeit (Schmettern).

Nach jeweils fünf Punkten wechselt das Aufgaberecht. Ist ein Satz beendet, erfolgt ein Seitenwechsel. Spieler 1 führt den braunen Schläger und verwendet den Joystick in Port 1. Der zweite Spieler steuert seinen (roten) Schläger mit dem Joystick in Port 2.

Anfangs ist es sicherlich etwas schwierig, beispielsweise das Anschneiden des Balls geschickt einzusetzen. Nach kurzer Zeit werden Sie den Ball immer besser unter Kontrolle haben und Ball und Gegner beherrschen.

Martin Jobst/kn

Eingabehinweise

Starten Sie zur Generierung des lauffähigen Programms Amiga-Basic, das sich auf der, jedem Amiga beiliegenden, »Extras«-Diskette befindet. Geben Sie im Direktmodus die Zeile

CLEAR,35000

ein, gefolgt von <RETURN>. Laden Sie anschließend »Ping Pong_Gen« und entfernen den Schreibschutz auf der Diskette im eingebauten Laufwerk. Starten Sie nun die Programmausführung mit RUN. Jetzt leuchtet die LED des Laufwerks df0., das Programm »Ping Pong« wird auf die eingegabte Diskette geschrieben. Nachdem sich Amiga-Basic mit »OK.« meldet, verlassen Sie mit QUIT den Basic-Interpreter. Das auf diese Weise erzeugte Programm »Ping Pong« kann von jetzt an vom CLI aus gestartet werden. Wie man Programme vom CLI oder das CLI selbst aufruft, steht in Ihrem Amiga-Handbuch.

Besitzer einer Speichererweiterung oder eines Amiga 2000 sollten beachten, daß »Ping Pong« nur ohne Zusatzspeicher lauffähig ist. Dieser muß also vor dem Start unbedingt mit »No-FastMem« oder hardwaremäßig abgeschaltet werden.

Simulation. Zwei Spieler treten dabei gegeneinander an.

```
110 B$ DATA 55,55,20,3c,00,00,00,0a,61,00
111 B$ DATA 14,20,91,fc,00,00,01,b6,51,c9
112 H$ DATA ff,ea,20,3c,00,00,00,1e,20,7a
113 V$ DATA 1b,b8,d1,fc,00,00,1f,be,3c,3c
114 G$ DATA c0,00,61,00,13,fc,22,3c,00,00
115 O$ DATA 00,0b,20,7a,1b,ae,d1,fc,00,00
116 Q$ DATA 1f,be,3c,3c,ff,ff,20,3c,00,00
117 C$ DATA 00,1e,61,00,13,e0,91,fc,00,00
118 W$ DATA 04,d6,51,c9,ff,ea,22,3c,00,00
119 F$ DATA 00,0b,20,7a,1b,80,d1,fc,00,00
120 U$ DATA 1f,be,3c,3c,ff,ff,20,3c,00,00
121 Y$ DATA 00,1e,61,00,13,b8,91,fc,00,00
122 A$ DATA 04,d6,51,c9,ff,ea,20,3c,00,00
123 O$ DATA 00,1e,20,7a,1b,58,d1,fc,00,00
124 A$ DATA 1f,d4,3c,3c,ff,fc,61,00,13,88
125 M$ DATA 22,3c,00,00,00,0b,20,7a,1b,38
126 T$ DATA d1,fc,00,00,1f,96,3c,3c,ff,ff
127 A$ DATA 20,3c,00,00,00,00,61,00,13,78
128 B$ DATA 91,fc,00,00,00,26,51,c9,ff,ea
129 W$ DATA 22,3c,00,00,00,0b,20,7a,1b,14
130 Z$ DATA d1,fc,00,00,24,96,3c,3c,ff,ff
131 S$ DATA 20,3c,00,00,00,00,61,00,13,50
132 O$ DATA 91,fc,00,00,00,26,51,c9,ff,ea
133 M$ DATA 22,3c,00,00,00,09,20,7a,1a,ee
134 Y$ DATA d1,fc,00,00,21,f0,3c,3c,ff,ff
135 L$ DATA 20,3c,00,00,00,01,61,00,13,28
136 F$ DATA 91,fc,00,00,00,4e,51,c9,ff,ea
137 J$ DATA 20,7a,1a,ea,d1,fc,00,00,20,88
138 K$ DATA 20,3c,00,00,00,06,22,7c,00,00
139 M$ DATA 1a,68,61,00,13,14,20,7a,1a,b0
140 S$ DATA d1,fc,00,00,22,90,20,3c,00,00
141 A$ DATA 00,06,22,7c,00,00,1a,68,61,00
142 C$ DATA 12,fa,20,7a,1a,96,d1,fc,00,00
143 G$ DATA 20,8e,20,3c,00,00,00,03,22,7c
144 I$ DATA 00,00,1a,92,61,00,13,06,20,7a
145 N$ DATA 1a,7c,d1,fc,00,00,22,96,20,3c
146 B$ DATA 00,00,00,03,22,7c,00,00,1a,9a
147 S$ DATA 61,00,12,ee,13,fc,00,15,00,00
148 M$ DATA 1f,90,61,00,0e,80,33,fc,01,2e
149 W$ DATA 00,00,1f,58,13,fc,00,44,00,00
150 G$ DATA 1f,81,33,fc,00,00,00,1f,5c
151 F$ DATA 13,fc,00,06,30,00,1f,60,2c,76
```

```
152 V$ DATA 00,04,4e,ae,ff,7c,4b,f9,00,df
153 E$ DATA f0,00,3b,7c,03,ef,00,96,2b,7a
154 L$ DATA 00,00,1a,f2,00,a0,2b,7c,00,00
155 X$ DATA 1a,e2,00,b0,2b,7c,02,1f,02,5f
156 S$ DATA 00,e4,3b,7c,00,08,00,b4,3b,7e
157 V$ DATA 00,40,00,ea,2b,7c,01,90,00,40
158 R$ DATA 00,b6,3b,7c,00,ff,00,9e,2b,7a
159 U$ DATA 00,00,1a,04,00,80,42,6a,00,88
160 V$ DATA 2b,7c,29,81,29,f2,00,8e,2b,7e
161 F$ DATA 00,38,00,d0,00,92,2b,7c,42,00
162 S$ DATA 00,00,01,00,3b,7c,00,17,01,04
163 M$ DATA 2b,7c,00,06,0f,ff,01,80,2b,7e
164 T$ DATA 00,00,00,90,01,84,2b,7c,07,77
165 J$ DATA 00,60,01,88,2b,7c,03,33,00,00
166 L$ DATA 01,8c,2b,7c,0f,ae,0f,00,01,e2
167 U$ DATA 3b,7c,0b,40,01,ae,2b,7c,0f,ff
168 S$ DATA 00,00,01,ea,2b,7c,0f,ae,06,00
169 W$ DATA 01,ba,3b,7c,0b,40,01,be,20,7e
170 Y$ DATA 00,df,f1,90,20,3c,00,00,00,07
171 D$ DATA 30,ba,00,00,d1,fc,00,00,00,02
172 P$ DATA 51,c8,ff,f4,2b,7c,00,0f,0f,ff
173 Q$ DATA 01,98,3b,7c,0f,f0,01,9c,3b,7c
174 Z$ DATA 83,a0,00,96,22,3a,19,78,10,3a
175 D$ DATA 19,9d,0c,39,00,01,00,00,1f,93
176 J$ DATA 67,00,00,0e,34,39,00,df,f0,0a
177 W$ DATA 4e,f9,00,00,06,04,34,39,00,df
178 G$ DATA f0,0c,08,02,00,01,67,00,00,1e
179 O$ DATA 48,41,0c,41,01,90,62,00,00,12
180 D$ DATA 48,41,06,81,00,00,55,56,23,c1
181 J$ DATA 00,00,1f,58,48,41,48,41,08,02
182 U$ DATA 00,09,67,00,00,1e,48,41,0e,41
183 O$ DATA 00,82,63,00,00,12,48,41,04,81
184 S$ DATA 00,00,55,56,23,c1,00,00,1f,58
185 P$ DATA 48,41,48,41,22,7c,00,00,1f,40
186 U$ DATA 24,7c,00,00,1f,44,12,80,14,80
187 J$ DATA 48,41,08,01,00,00,67,00,00,14
188 A$ DATA 13,7c,00,00,00,03,15,7c,00,01
189 S$ DATA 00,03,4e,f9,00,00,06,84,13,7c
190 Z$ DATA 00,00,00,03,15,7c,00,00,00,03
191 S$ DATA e2,f9,13,41,00,01,06,01,00,08
192 A$ DATA 15,41,00,01,06,00,00,11,13,40
193 X$ DATA 00,02,04,00,00,01,15,40,00,02
```

```
194 V$ DATA 22,3a,18,b8,10,3e,18,d8,0c,39
195 X$ DATA 00,01,00,00,1f,93,67,00,00,0e
196 W$ DATA 34,39,00,df,f0,0c,4e,f9,00,00
197 Z$ DATA 06,c8,34,39,00,df,f0,0a,08,02
198 Y$ DATA 00,01,67,00,00,1e,48,41,0a,41
199 G$ DATA 01,90,62,00,00,12,48,41,06,81
200 A$ DATA 00,00,55,56,23,c1,00,00,1f,5c
201 H$ DATA 48,41,48,41,08,02,00,09,67,00
202 S$ DATA 00,1e,48,41,0c,41,08,82,63,00
203 I$ DATA 00,12,48,41,04,81,00,00,55,56
204 G$ DATA 23,c1,00,00,1f,5c,48,41,48,41
205 J$ DATA 22,7c,00,00,1f,50,24,7c,00,00
206 J$ DATA 1f,54,12,80,14,80,48,41,08,01
207 O$ DATA 00,00,67,00,00,14,13,7c,00,01
208 C$ DATA 00,03,15,7c,00,01,00,03,4e,f9
209 I$ DATA 00,00,07,48,13,7c,00,00,00,03
210 N$ DATA 15,7c,00,00,00,03,e2,59,13,41
211 G$ DATA 00,01,50,01,15,41,00,01,06,00
212 F$ DATA 00,11,13,40,00,02,15,40,00,02
213 K$ DATA 0e,39,00,01,00,00,1f,83,67,00
214 A$ DATA 00,c6,0c,39,00,01,00,00,1f,8d
215 L$ DATA 66,00,00,32,0c,39,00,01,00,00
216 B$ DATA 1f,93,67,00,00,14,08,39,00,07
217 L$ DATA 00,bf,e0,01,66,00,00,68,4e,f9
218 D$ DATA 00,00,07,d2,08,39,00,06,00,bf
219 W$ DATA e0,01,66,00,00,56,4e,f9,00,00
220 R$ DATA 07,d2,0c,39,00,01,00,00,1f,93
221 T$ DATA 67,00,00,14,08,39,00,06,00,bf
222 C$ DATA e0,01,66,00,00,38,4e,f9,00,00
223 A$ DATA 07,d2,08,39,00,07,00,bf,e0,01
224 I$ DATA 66,00,00,26,23,fc,00,00,14,00
225 E$ DATA 00,00,1f,74,23,fc,00,20,00,00
226 H$ DATA 00,00,1f,78,13,fc,00,01,00,00
227 V$ DATA 1f,83,13,fc,00,01,00,00,1f,82
228 T$ DATA 0e,39,00,01,00,00,1f,8d,66,00
229 U$ DATA 00,1e,36,3a,17,58,06,43,00,12
230 Z$ DATA 61,00,0f,1e,33,fc,45,00,00,00
```

Listing 1. Geben Sie »PingPong_Gen« bitte mit dem Checksummer (Seite 143) ein

231 on DATA 1f,72,4e,f9,00,00,08,30,36,3a
 232 3K DATA 17,3a,06,43,00,09,61,00,0f,04
 233 yS DATA 33,fe,55,00,00,00,1f,72,20,3a
 234 ng DATA 17,42,91,b9,00,00,1f,78,20,3a
 235 QS DATA 17,3a,d1,b9,00,00,1f,7c,30,3a
 236 DK DATA 17,1e,d1,79,00,00,1f,72,18,3a
 237 uV DATA 17,22,20,3a,17,0c,d1,b9,00,00
 238 pI DATA 1f,66,20,3a,17,08,d1,b9,00,00
 239 qh DATA 1f,6a,36,3a,17,06,0c,39,00,46
 240 fF DATA 00,00,1f,72,62,00,00,12,13,fe
 241 J9 DATA 00,01,00,00,1f,86,13,fe,00,01
 242 GN DATA 00,00,1f,8b,0c,39,00,eb,00,00
 243 QJ DATA 1f,72,63,00,00,12,13,fe,00,01
 244 5b DATA 00,00,1f,86,13,fe,00,02,00,00
 245 vM DATA 1f,8b,0c,39,00,3c,00,00,1f,7c
 246 lw DATA 63,00,01,1a,0c,39,00,01,00,00
 247 yC DATA 1f,82,66,00,00,2a,0c,39,00,55
 248 GU DATA 00,00,1f,72,66,00,00,0a,13,fe
 249 Vd DATA 00,01,00,00,1f,86,0c,39,00,45
 250 BW DATA 00,00,1f,72,66,00,00,0a,13,fe
 251 U8 DATA 00,01,00,00,1f,86,23,fe,00,1a
 252 SN DATA 00,00,00,00,1f,78,23,fe,00,00
 253 5T DATA 18,00,00,00,1f,74,0c,39,00,01
 254 es DATA 00,00,1f,8c,66,00,00,0a,13,fe
 255 DE DATA 00,01,00,00,1f,86,0c,00,04,66
 256 JU DATA 65,00,00,12,13,fe,00,01,00,00
 257 og DATA 1f,86,13,fe,00,02,00,00,1f,8b
 258 7W DATA 0c,04,00,33,62,00,00,12,13,fe
 259 bR DATA 00,01,00,00,1f,86,13,fe,00,01
 260 4I DATA 00,00,1f,8b,3a,3c,00,00,3c,3c
 261 IK DATA 00,46,9c,04,04,46,00,05,0c,46
 262 QM DATA 00,81,62,00,00,08,52,45,4e,fa
 263 29 DATA ff,f0,3c,3c,00,00,b0,dc,45,bc,43
 264 rr DATA 65,00,00,0a,13,fe,00,01,00,00
 265 Wa DATA 1f,86,3c,3c,01,6e,9c,45,bc,43
 266 kh DATA 62,00,00,0a,13,fe,00,01,00,00
 267 ta DATA 1f,86,0c,39,00,00,00,1f,86
 268 OH DATA 66,00,00,3e,13,fe,00,01,00,00
 269 pe DATA 1f,8a,33,fe,01,90,00,df,f0,a6
 270 nJ DATA 33,fe,82,01,00,df,f0,96,13,fe
 271 ch DATA 00,14,00,00,1f,89,0c,39,00,00
 272 gH DATA 00,00,1f,82,67,00,00,12,13,fe
 273 jJ DATA 00,00,00,00,1f,8a,13,fe,00,00
 274 ie DATA 00,00,1f,82,04,39,00,01,00,00
 275 Y6 DATA 1f,89,0c,39,00,00,00,00,1f,89
 276 wh DATA 66,00,00,0a,33,fe,00,01,00,df
 277 H5 DATA 0c,96,0c,39,00,93,00,00,1f,72
 278 87 DATA 67,00,00,14,0c,39,00,92,00,00
 279 go DATA 1f,72,67,00,00,08,4e,f9,00,00
 280 eZ DATA 0a,38,0c,39,00,0c,00,00,1f,7c
 281 aW DATA 62,00,00,2e,3a,3c,00,00,9a,79
 282 fS DATA 00,00,1f,64,33,65,00,00,1f,64
 283 sp DATA 23,fe,00,00,30,00,00,00,1f,74
 284 71 DATA 23,fe,00,10,00,00,00,00,1f,78
 285 xI DATA 13,fe,00,01,00,00,1f,8c,13,fe
 286 nV DATA 00,ff,00,00,1f,85,1a,04,9a,3a
 287 14 DATA 15,38,13,fe,00,01,00,00,1f,84
 288 SE DATA 1c,05,9c,39,00,00,1f,80,20,3c
 289 kW DATA 00,00,00,04,1e,3c,00,00,bc,07
 290 T4 DATA 66,00,00,08,13,67,00,00,1f,85
 291 Jg DATA 52,07,51,68,ff,f0,20,3c,00,00
 292 Ap DATA 00,04,1e,3c,00,0c,bc,07,66,00
 293 ot DATA 00,08,13,60,00,00,1f,85,52,07
 294 B4 DATA 51,68,ff,f0,20,3c,00,00,00,04
 295 bM DATA 1e,3c,00,06,bc,07,66,00,00,0a
 296 Z8 DATA 13,fe,00,06,00,00,1f,85,52,07
 297 bM DATA 51,68,ff,ee,20,3c,00,00,00,01
 298 Pb DATA 1e,3c,00,04,bc,07,66,00,00,0a
 299 Y5 DATA 13,fe,00,04,00,00,1f,85,52,07
 300 FF DATA bc,07,66,00,00,0a,13,fe,00,04
 301 5q DATA 00,00,1f,85,5a,07,51,68,ff,ee
 302 ak DATA 3a,3a,14,92,9a,7a,14,7c,0c,39
 303 KU DATA 00,00,00,00,1f,85,66,00,00,24
 304 vP DATA 0c,06,00,0c,65,00,00,1c,0c,06
 305 P5 DATA 00,15,62,00,00,14,13,fe,00,00
 306 xQ DATA 00,00,1f,84,23,fe,00,00,00,00
 307 Mw DATA 00,00,1f,6a,0c,39,00,01,00,00
 308 R6 DATA 1f,85,66,00,0c,3e,0c,06,00,0b
 309 WZ DATA 65,00,00,36,0c,06,00,18,62,00
 310 5M DATA 00,2e,13,fe,00,00,00,00,1f,84
 311 Vn DATA 0c,06,00,0c,62,00,00,0c,23,fe
 312 3u DATA ff,ff,f9,00,00,00,1f,6a,0c,06
 313 9Z DATA 00,16,65,00,00,0c,23,fe,00,00

314 11 DATA 07,00,30,00,1f,6a,0c,39,00,02
 315 7n DATA 00,00,1f,85,66,00,00,3e,0c,06
 316 xL DATA 00,0a,65,00,00,36,0c,06,00,19
 317 7t DATA 62,00,00,2e,13,fe,00,00,00,00
 318 4b DATA 1f,84,0c,06,00,0c,62,00,00,0c
 319 9T DATA 23,fe,ff,ff,f9,00,00,00,1f,6a
 320 3e DATA 0c,06,00,16,65,00,00,23,fe
 321 xL DATA 00,00,07,00,00,00,1f,6a,0c,39
 322 7t DATA 00,03,00,00,1f,85,66,00,00,50
 323 3e DATA 0c,06,00,0a,65,00,00,48,0c,06
 324 7t DATA 00,1b,62,00,00,40,13,fe,00,00
 325 3e DATA 00,00,1f,84,0c,06,00,0c,62,00
 326 3e DATA 00,0c,23,fe,ff,ff,f9,00,00,00
 327 7t DATA 1f,6a,0c,06,00,16,65,00,00,0c
 328 3e DATA 23,fe,00,00,07,00,00,00,1f,6a
 329 7t DATA 0c,06,00,1a,65,00,00,0c,23,fe
 330 3e DATA 00,00,0c,23,fe,ff,ff,f9,00,00,00
 331 2v DATA 00,00,00,1f,85,66,00,00,50
 332 3e DATA 0c,06,00,1a,65,00,00,0c,23,fe
 333 7t DATA 00,1c,62,00,00,40,13,fe,00,00
 334 7t DATA 00,00,1f,84,0c,06,00,0c,62,00
 335 3e DATA 00,0c,23,fe,ff,ff,f9,00,00,00
 336 7t DATA 1f,6a,0c,06,00,16,65,00,00,0c
 337 3e DATA 23,fe,00,00,07,00,00,00,1f,6a
 338 7t DATA 0c,06,00,1a,65,00,00,0c,23,fe
 339 3e DATA 00,00,0c,23,fe,ff,ff,f9,00,00,00
 340 7t DATA 00,06,00,00,1f,85,66,00,00,50
 341 x DATA 0c,06,00,0a,65,00,00,48,0c,06
 342 7t DATA 00,1c,62,00,00,40,13,fe,00,00
 343 7t DATA 00,00,1f,84,0c,06,00,0c,62,00
 344 7t DATA 00,0c,23,fe,ff,ff,f9,00,00,00
 345 7t DATA 1f,6a,0c,06,00,16,65,00,00,0c
 346 7t DATA 23,fe,00,00,07,00,00,00,1f,6a
 347 7t DATA 0c,06,00,1a,65,00,00,0c,23,fe
 348 7t DATA 00,00,0c,23,fe,ff,ff,f9,00,00,00
 349 xJ DATA 00,45,00,00,1f,72,67,00,00,36
 350 7t DATA 0c,39,00,d4,00,00,1f,72,67,00
 351 7t DATA 00,3a,0c,39,00,46,00,00,1f,72
 352 7t DATA 67,00,00,2e,0c,39,00,43,00,00
 353 7t DATA 1f,72,67,00,00,22,0c,39,00,d2
 354 7t DATA 00,00,1f,72,67,00,0c,16,0c,39
 355 7t DATA 00,87,00,00,1f,72,67,00,00,0a
 356 7t DATA 13,fe,00,01,00,00,1f,84,0c,39
 357 7t DATA 00,01,00,00,1f,84,67,00,00,0c
 358 39 DATA 0c,39,00,01,00,00,1f,93,67,00
 359 7t DATA 00,14,08,39,00,07,00,bf,e0,01
 360 7t DATA 66,00,00,b4,4e,f9,00,00,04,34
 361 7t DATA 08,39,00,06,00,bf,e0,01,66,00
 362 7t DATA 00,a2,33,fe,ff,df,00,00,1f,64
 363 7t DATA 61,00,09,aa,0c,39,00,01,00,00
 364 7t DATA 1f,93,67,00,00,0c,34,39,00,df
 365 7t DATA f0,0c,4e,f9,00,00,0d,5e,34,39
 366 7t DATA 00,df,f0,0a,08,02,00,01,67,00
 367 7t DATA 00,0c,23,fe,ff,ff,ff,ff,ff,ff,ff
 368 MN DATA 1f,60,08,02,00,09,67,00,00,0c
 369 7t DATA 23,fe,00,00,00,01,00,00,1f,60
 370 7t DATA 0c,39,00,01,00,00,1f,84,67,00
 371 7t DATA 00,1e,0c,39,00,01,00,00,1f,82
 372 7t DATA 66,00,00,12,23,fe,00,00,00,00
 373 7t DATA 00,00,1f,78,4e,f9,00,00,0d,44
 374 7t DATA 3c,02,e2,4e,b5,46,08,06,00,08
 375 7t DATA 67,00,00,1e,33,fe,ff,e4,00,00
 376 7t DATA 1f,64,23,fe,00,20,00,00,00,00
 377 7t DATA 1f,78,23,fe,00,00,30,00,00,00
 378 7t DATA 1f,74,13,fe,00,01,00,00,1f,84
 379 7t DATA 13,fe,00,ff,00,00,1f,85,1c,03
 380 7t DATA 9c,3a,11,99,20,3c,00,00,00,04
 381 7t DATA 1e,3c,00,00,bc,07,66,00,00,08
 382 7t DATA 13,67,00,00,1f,85,52,07,51,68
 383 7t DATA ff,f0,20,3c,00,00,00,04,1e,3c
 384 7t DATA 00,0c,bc,07,66,00,00,08,13,00
 385 7t DATA 00,00,1f,85,52,07,51,68,ff,f0
 386 7t DATA 20,3c,00,00,00,04,1e,3c,00,06
 387 7t DATA bc,07,66,00,00,0a,13,fe,00,06
 388 7t DATA 00,00,1f,85,52,07,51,68,ff,ee
 389 7t DATA 20,3c,00,00,00,01,1e,3c,00,04
 390 7t DATA bc,07,66,00,00,0a,13,fe,00,04
 391 7t DATA 00,00,1f,85,52,07,bc,07,66,00
 392 7t DATA 00,0a,13,fe,00,04,00,00,1f,85
 393 7t DATA 1f,85,66,00,00,00,00,00,00
 394 7t DATA 1f,85,66,00,00,00,00,00,00
 395 7t DATA 1f,85,66,00,00,00,00,00,00
 396 7t DATA 65,00,00,1c,0c,06,00,0c,62,00

397 f5 DATA 00,14,13,fe,00,00,00,00,1f,84
 398 61 DATA 23,fe,00,00,00,00,00,00,1f,6a
 399 7t DATA 0c,39,00,01,00,00,1f,85,66,00
 400 7t DATA 00,3e,0c,06,00,04,65,00,00,36
 401 7t DATA 0c,06,00,11,62,00,00,2e,13,fe
 402 7t DATA 00,00,00,00,1f,84,0c,06,00,07
 403 7t DATA 62,00,00,0c,23,fe,ff,ff,f9,00
 404 7t DATA 00,00,1f,6a,0c,06,00,0c,65,00
 405 7t DATA 00,0c,23,fe,00,00,07,00,00,00
 406 7t DATA 1f,6a,0c,39,00,02,00,00,1f,85
 407 7t DATA 66,00,00,3a,0c,06,00,03,65,00
 408 7t DATA 00,36,0c,06,00,12,62,00,00,2e
 409 7t DATA 13,fe,00,00,00,00,1f,84,0c,06
 410 7t DATA 00,07,62,00,00,0c,23,fe,ff,ff
 411 7t DATA f9,00,00,00,1f,6a,0c,06,00,0c
 412 7t DATA 65,00,00,0c,23,fe,00,00,07,00
 413 7t DATA 00,00,1f,6a,0c,39,00,03,00,00
 414 7t DATA 1f,85,66,00,00,50,0c,06,00,01
 415 7t DATA 65,00,00,48,0c,06,00,12,62,00
 416 7t DATA 00,40,13,fe,00,00,00,00,1f,84
 417 7t DATA 0c,06,00,07,62,00,00,0c,23,fe
 418 7t DATA ff,ff,f9,00,00,00,1f,6a,0c,06
 419 7t DATA 00,03,62,00,00,0c,23,fe,ff,ff
 420 7t DATA f2,00,00,00,1f,6a,0c,06,00,0c
 421 7t DATA 65,00,00,0c,23,fe,00,00,07,00
 422 7t DATA 00,00,1f,6a,0c,39,00,04,00,00
 423 7t DATA 1f,85,66,00,00,50,0c,06,00,00
 424 7t DATA 65,00,00,48,0c,06,00,12,62,00
 425 7t DATA 00,40,13,fe,00,00,00,00,1f,84
 426 7t DATA 0c,06,00,07,62,00,00,0c,23,fe
 427 7t DATA ff,ff,f2,00,00,00,1f,6a,0c,06
 428 7t DATA 00,03,62,00,00,0c,23,fe,ff,ff
 429 7t DATA f2,00,00,00,1f,6a,0c,06,00,0c
 430 7t DATA 65,00,00,0c,23,fe,00,00,07,00
 431 7t DATA 00,00,1f,6a,0c,39,00,06,00,00
 432 7t DATA 1f,85,66,00,00,50,0c,06,00,00
 433 7t DATA 65,00,00,48,0c,06,00,12,62,00
 434 7t DATA 00,40,13,fe,00,00,00,00,1f,84
 435 7t DATA 0a,06,00,07,62,00,00,0c,23,fe
 436 7t DATA ff,ff,f9,00,00,00,1f,6a,0c,06
 437 7t DATA 00,03,62,00,00,0c,23,fe,ff,ff
 438 7t DATA f2,00,00,00,1f,6a,0c,06,00,0c
 439 7t DATA 65,00,00,0c,23,fe,00,00,07,00
 440 7t DATA 00,00,1f,6a,0c,39,00,53,00,00
 441 7t DATA 1f,72,67,00,00,46,0c,39,00,54
 442 7t DATA 00,00,1f,72,67,00,00,3a,0c,39
 443 7t DATA 00,56,00,00,1f,72,67,00,00,2e
 444 7t DATA 0c,39,00,57,00,00,1f,72,67,00
 445 7t DATA 00,22,0c,39,00,58,00,00,1f,72
 446 7t DATA 67,00,00,16,0c,39,00,53,00,00
 447 7t DATA 1f,72,67,00,00,0a,13,fe,00,01
 448 7t DATA 00,00,1f,84,0c,39,00,01,00,00
 449 7t DATA 1f,84,67,00,00,0c,46,0c,39,00,01
 450 7t DATA 00,00,1f,93,67,00,00,14,08,39
 451 7t DATA 00,06,00,bf,e0,01,66,00,00,bc
 452 7t DATA 4e,f9,00,00,10,0c,08,39,00,07
 453 7t DATA 00,bf,e0,01,66,00,00,ea,33,fe
 454 7t DATA 00,23,00,00,1f,64,61,00,06,1e
 455 7t DATA 0c,39,00,01,00,00,1f,93,67,00
 456 7t DATA 00,0c,34,39,00,df,f0,0c,4e,f9
 457 7t DATA 00,00,10,22,34,39,00,df,f0,0c
 458 7t DATA 08,02,00,01,67,00,00,0c,23,fe
 459 7t DATA ff,ff,ff,ff,ff,ff,ff,ff,ff,ff
 460 7t DATA 00,09,67,00,00,0c,23,fe,00,00
 461 7t DATA 00,01,00,00,1f,60,0c,39,00,01
 462 7t DATA 00,01,00,00,1f,82,66,00,00,1c
 463 7t DATA 23,fe,00,00,00,00,00,00,1f,78
 464 7t DATA 23,fe,00,00,30,00,00,00,1f,74
 465 7t DATA 4e,f9,00,00,11,72,3c,02,e2,4e
 466 7t DATA b5,46,08,06,00,00,67,00,00,1e
 467 7t DATA 33,fe,00,3c,00,00,1f,64,23,fe
 468 7t DATA 00,20,00,00,00,00,1f,78,23,fe
 469 7t DATA 00,00,30,00,00,00,1f,74,0c,39
 470 7t DATA 00,01,00,00,1f,85,66,00,01,08
 471 7t DATA 0c,39,00,01,00,00,1f,8c,66,00
 472 7t DATA 00,26,0c,39,00,00,00,00,1f,72
 473 7t DATA 62,00,00,0c,52,39,00,00,1f,88
 474 7t DATA 4e,f9,00,00,12,74,52,39,00,00
 475 7t DATA 1f,87,4e,f9,00,00,12,74,0c,39
 476 7t DATA 00,01,00,00,00,1f,8a,66,00,00,32
 477 7t DATA 0c,39,00,01,00,00,1f,8b,66,00
 478 7t DATA 00,0c,52,39,00,00,1f,88,4e,f9


```

480 VT DATA 00,00,12,74,0a,39,00,32,00,00
481 Sh DATA 1f,8b,66,00,00,0e,52,39,00,00
482 Jn DATA 1f,87,4e,f9,00,00,12,74,0e,39
483 QK DATA 00,00,00,00,1f,8a,66,00,00,00
484 LK DATA 0c,39,00,01,00,00,1f,82,66,00
485 K9 DATA 00,26,0e,39,00,93,00,00,1f,72
486 rI DATA 62,00,00,0e,52,39,00,00,1f,88
487 ka DATA 4e,f9,00,00,12,74,52,39,00,00
488 Pt DATA 1f,87,4e,f9,00,00,12,74,0e,39
489 bf DATA 00,93,00,00,1f,72,62,00,00,0e
490 wZ DATA 52,39,00,00,1f,87,4e,f9,00,00
491 T6 DATA 12,74,52,39,00,00,1f,88,4e,f9
492 da DATA 00,00,12,74,0e,39,00,01,00,00
493 et DATA 1f,87,66,00,00,0e,52,39,00,00
494 yE DATA 1f,87,4e,f9,01,00,12,74,0e,39
495 zm DATA 00,02,00,00,1f,8b,66,00,00,08
496 hI DATA 52,39,00,00,1f,88,61,00,00,0e
497 aT DATA 0c,39,00,01,00,00,1f,94,67,00
498 s8 DATA 00,8a,22,7e,00,00,1f,48,24,7e
499 pI DATA 00,00,1f,4e,08,03,00,00,67,00
500 rD DATA 00,14,13,7e,00,01,00,03,15,7e
501 vA DATA 00,01,00,03,0e,f9,00,00,12,74,0e,39
502 7w DATA 13,7e,00,00,00,03,15,7e,00,00
503 pD DATA 00,03,0e,4b,14,8a,54,04,15,44
504 EB DATA 00,02,15,43,00,01,12,85,54,05
505 pf DATA 13,45,00,02,13,43,00,01,33,7e
506 rC DATA 80,20,00,dF,f0,9c,23,fa,0c,66
507 ks DATA 00,00,19,70,23,fa,0c,62,00,00
508 uA DATA 19,b8,23,fa,0c,5a,00,00,19,58
509 bC DATA 23,fa,0c,5a,00,00,19,64,23,fa
510 mS DATA 0c,56,00,00,08,08,23,fa,0c,52
511 LR DATA 00,00,19,10,4e,fa,f2,d4,2a,7e
512 qF DATA 00,00,00,14,61,00,03,c2,51,cd
513 SK DATA ff,fa,61,00,03,54,61,00,02,f8
514 bg DATA 13,7e,00,00,00,0c,1f,8d,13,7e
515 lA DATA 00,00,00,00,00,1f,13,7e,00,00
516 Wm DATA 00,00,1f,92,61,00,00,1e,13,7e
517 eW DATA 00,00,00,00,1f,96,13,7e,00,00
518 Fh DATA 00,00,1f,94,13,7e,00,00,00,00
519 jn DATA 1f,95,4e,fa,f2,86,24,3e,00,00
520 VT DATA 00,00,14,39,70,00,1f,87,4e,39
521 P6 DATA 00,00,1f,88,84,7e,00,05,48,42
522 Hc DATA 0c,42,00,00,00,66,00,00,16,0c,39
523 bL DATA 00,01,00,00,1f,95,67,00,00,0a
524 tK DATA 08,79,00,00,00,00,1f,84,0c,39
525 IM DATA 00,01,00,00,1f,93,67,00,00,18
526 Tr DATA 13,fa,0b,eb,00,00,1f,8e,13,fa
527 rJ DATA 0b,04,00,00,1f,8f,4e,f9,00,00
528 wZ DATA 13,0e,13,fa,0b,d5,00,00,1f,8f
529 MI DATA 13,fa,0b,eb,00,00,1f,8e,0c,39
530 oc DATA 00,15,00,00,1f,87,65,00,00,18
531 FK DATA 08,79,00,00,00,00,1f,8d,13,7e
532 HF DATA 00,00,00,00,1f,95,4e,f9,00,00
533 F7 DATA 13,fa,0c,39,00,15,00,00,1f,88
534 wI DATA 65,00,00,12,08,79,00,00,00,00
535 wJ DATA 1f,8d,13,7e,00,01,00,00,1f,95
536 dP DATA 24,3e,00,00,00,00,14,3a,0b,88
537 sI DATA b4,3a,0b,86,65,00,00,2e,94,39
538 wr DATA 00,00,1f,8f,0c,72,00,01,66,00
539 pI DATA 00,0e,52,39,00,00,1f,90,4e,f9
540 Ge DATA 00,00,14,3e,08,79,00,00,00,00
541 lF DATA 1f,93,52,39,00,00,1f,91,61,00
542 xb DATA 02,3e,14,3a,0b,51,b4,3a,0b,4e
543 Yz DATA 65,00,00,30,94,39,00,00,1f,8e
544 P9 DATA 0c,02,00,01,66,00,00,0e,52,39
545 uA DATA 00,00,1f,90,4e,f9,00,00,14,7e
546 gc DATA 08,79,00,00,00,00,1f,93,0c,39
547 Ww DATA 00,01,70,00,1f,92,61,00,01,7e
548 z3 DATA 0c,39,00,02,00,00,1f,91,66,00
549 V2 DATA 00,0a,13,7e,00,01,00,00,1f,94
550 AA DATA 0c,39,00,02,00,00,1f,92,66,00
551 X4 DATA 00,0a,13,7e,00,01,00,00,1f,94
552 Jw DATA 14,39,00,00,1f,91,44,39,00,00
553 e3 DATA 1f,92,0c,02,00,00,02,66,00,0e
554 YI DATA 0c,39,00,01,00,00,1f,96,67,00
555 Bn DATA 00,22,0e,39,00,0a,00,00,1f,8e
556 bI DATA 66,00,00,06,61,00,01,de,0c,39
557 ZM DATA 00,0a,00,00,1f,8f,66,00,00,06
558 Fe DATA 61,00,01,0c,0c,39,00,01,30,00
559 tI DATA 1f,93,67,00,00,18,33,7e,0f,70
560 wS DATA 0c,dF,f1,84,33,7e,06,00,70,dF
561 K2 DATA f1,bc,4e,f9,00,00,15,10,33,7e
562 eS DATA 06,00,00,dF,f1,84,33,7e,0c,00

```

```

563 OQ DATA 00,dF,f1,bc,14,3a,0a,7c,84,7e
564 FV DATA 00,0a,20,7e,0a,16,d1,7c,30,00
565 wa DATA 20,92,20,3e,00,00,00,03,61,00
566 tO DATA 02,dc,61,00,02,b1,48,42,4e,f9
567 oC DATA 00,00,15,3e,14,3a,0a,54,20,7a
568 xH DATA 09,f2,11,7e,00,00,20,94,20,3e
569 JL DATA 00,00,00,03,61,00,02,b8,61,00
570 Ja DATA 02,68,24,3e,00,00,00,00,14,3a
571 MJ DATA 0a,33,84,7e,00,0a,20,7a,09,0e
572 nC DATA d1,7e,00,00,22,9a,20,3e,00,00
573 7n DATA 00,03,61,00,02,92,61,00,02,68
574 sr DATA 48,42,4e,f9,00,00,15,86,14,3a
575 sS DATA 0a,0b,20,7a,09,a8,d1,7e,00,00
576 xT DATA 22,9a,20,3e,00,00,03,61,00
577 zG DATA 02,6e,61,00,02,1e,20,7a,09,94
578 ke DATA d1,7e,00,00,20,98,20,3e,00,00
579 WY DATA 00,03,14,39,00,00,1f,91,61,00
580 yI DATA 02,50,61,00,02,26,20,7a,09,7e
581 rJ DATA d1,7e,00,00,22,a0,20,3e,00,00
582 JI DATA 00,03,14,39,00,00,1f,92,61,00
583 QH DATA 02,32,61,00,02,08,23,7e,00,00
584 7D DATA 00,00,00,00,19,58,23,7e,00,00
585 JI DATA 70,00,70,00,19,64,33,7e,82,02
586 lA DATA 00,dF,f0,96,3b,7c,01,40,00,b6
587 Kh DATA 61,00,00,d8,3b,7c,01,ab,00,b6
588 rK DATA 61,00,00,0c,33,7e,00,02,00,dF
589 oQ DATA f0,96,61,00,00,04,4e,75,13,7e
590 rQ DATA 00,01,00,00,1f,62,13,7e,00,00
591 St DATA 00,00,1f,83,13,7e,00,00,00,00
592 at DATA 1f,86,23,7e,00,00,00,00,00,00
593 wk DATA 1f,74,23,7e,00,00,00,00,00,00
594 77 DATA 1f,60,33,7e,00,00,00,00,1f,64
595 oJ DATA 23,7e,00,00,00,00,00,00,1f,66
596 JI DATA 13,7e,00,00,00,00,1f,8a,13,7e
597 oB DATA 00,00,00,00,1f,8b,13,7e,00,00
598 x3 DATA 00,00,1f,8c,4e,75,13,7e,00,01
599 AC DATA 00,00,1f,8d,13,7e,00,00,00,00
600 lI DATA 1f,75,13,7e,00,00,15,30,00,1f,90
601 BS DATA 13,7e,00,00,00,00,1f,87,13,7e
602 Jy DATA 00,00,00,00,1f,88,13,7e,00,00
603 JV DATA 00,00,1f,8e,13,7e,00,00,00,00
604 x3 DATA 1f,8f,4e,75,08,79,00,00,00,00
605 rJ DATA 1f,93,13,7e,00,00,01,00,00,1f,96
606 EO DATA 14,3a,08,eb,13,fa,08,eb,00,00
607 eS DATA 1f,87,13,7e,00,00,1f,88,79,79
608 kv DATA 00,00,00,00,1f,8d,4e,75,24,3e
609 Pa DATA 00,00,00,01,51,08,ff,7e,20,3e
610 rO DATA ff,ff,ff,ff,51,ca,ff,f4,4e,75
611 x4 DATA 13,7e,00,00,00,00,1f,8a,33,7e
612 5K DATA 02,5f,00,dF,f0,96,33,7e,82,01
613 wZ DATA 00,dF,f0,96,13,7e,00,3e,00,00
614 vA DATA 1f,89,23,7e,00,1a,00,00,00,00
615 wA DATA 1f,78,23,7e,00,00,18,00,00,00
616 VD DATA 1f,74,23,fa,08,48,00,00,1f,66
617 7T DATA 4e,75,33,7e,00,00,1f,68,23,7e
618 aT DATA 13,00,00,00,00,00,1f,7e,20,3e
619 rF DATA 00,00,00,00,4e,75,30,86,d1,7e
620 JN DATA 00,00,00,28,51,08,ff,f6,20,3e
621 Jc DATA 00,00,00,04,e3,5e,51,09,ff,ea
622 7b DATA 4e,75,30,86,41,7e,00,00,28
623 gr DATA 51,05,ff,f6,20,3e,00,00,00,04
624 JH DATA e2,5e,51,09,ff,ea,4e,75,30,86
625 y9 DATA d1,7e,00,00,00,28,51,08,ff,f6
626 Ne DATA 4e,75,30,86,d1,7e,00,00,28
627 Nb DATA e3,5e,51,08,ff,f4,4e,75,24,11
628 XT DATA 20,82,d3,7e,00,00,00,04,d1,7e
629 MQ DATA 00,00,00,04,34,11,30,82,d3,7e
630 KC DATA 00,00,00,02,d1,7e,00,00,00,24
631 dI DATA 51,08,ff,f6,4e,75,ca,38,70,0f
632 rR DATA 14,11,10,82,d1,7e,00,00,00,28
633 Jd DATA 14,29,00,01,10,82,d3,7e,00,00
634 w4 DATA 00,02,d1,7e,00,00,00,28,51,08
635 xJ DATA ff,de,4e,75,14,11,11,22,00,01
636 zI DATA d1,7e,00,00,00,28,14,29,00,01
637 sJ DATA 11,42,00,01,d3,7e,00,00,00,02
638 qK DATA d1,7e,00,00,00,28,51,08,ff,f6
639 7m DATA 4e,75,00,00,00,00,66,00,00,08
640 xK DATA 22,7e,00,00,1a,da,0e,02,00,01
641 Uy DATA 66,00,00,08,22,7e,00,00,1a,92
642 IR DATA 0c,02,00,02,66,00,00,08,22,7e
643 D6 DATA 00,00,1a,9a,0c,02,00,03,66,00
644 qY DATA 00,08,22,7e,00,00,1a,22,0c,02
645 qO DATA 00,04,66,00,30,06,22,7e,00,00

```

```

646 pI DATA 1a,aa,0c,02,00,05,66,00,00,08
647 ND DATA 22,7c,07,00,1a,b2,0c,02,00,06
648 2w DATA 66,00,00,08,22,7e,00,00,1a,ba
649 Zn DATA 0c,02,00,07,66,00,00,08,22,7e
650 Ch DATA 00,00,1a,e2,0c,02,00,08,66,00
651 5a DATA 00,08,22,7e,00,00,1a,ca,0c,02
652 RH DATA 00,00,00,00,08,22,7e,00,00
653 CZ DATA 1a,d2,4e,75,22,3e,00,00,00,07
654 Q6 DATA d1,7e,00,00,00,01,1e,10,08,06
655 R9 DATA 00,07,67,00,00,06,08,43,00,00
656 dB DATA 08,03,00,00,67,00,00,06,08,06
657 pr DATA 00,07,e3,1e,51,c9,ff,e4,10,86
658 4g DATA 51,08,ff,d0,4e,75,a0,5a,b1,00
659 9I DATA 00,00,00,01,70,00,00,0f,00,00
660 2a DATA 00,1f,00,00,00,7f,00,80,00,7f
661 Np DATA 00,80,00,7f,01,80,00,7f,ff,80
662 JP DATA 30,7f,ff,80,ff,ff,ff,80,00,7f
663 FT DATA 31,80,00,7f,00,80,00,7f,0c,80
664 lR DATA 00,7f,00,00,00,7f,00,00,00,1f
665 YP DATA 00,00,00,00,00,00,01,a0,5a
666 Um DATA b1,00,00,00,7e,00,00,00,ff,80
667 2x DATA 00,00,ff,e0,00,00,ff,00,00,00
668 ny DATA ff,f8,00,00,ff,f8,00,00,ff,f8
669 FF DATA 00,00,ff,f8,00,00,ff,f8,00,00
670 bO DATA ff,f8,00,00,ff,f8,00,00,ff,f8
671 bV DATA 00,00,ff,f8,00,00,ff,f8,00,00
672 HJ DATA ff,e0,00,00,ff,80,00,00,7e,00
673 OJ DATA a0,6a,a2,00,e0,00,00,00,00,00
674 yq DATA 00,00,a8,6a,aa,00,00,00,00,00
675 BF DATA 00,00,c0,00,a0,5a,b1,00,00,00
676 t4 DATA 00,fe,00,00,07,ff,00,00,0f,ff
677 PL DATA 00,00,3f,ff,00,00,7f,ff,00,00
678 Mb DATA 7f,ff,00,70,ff,ff,0c,00,ff,ff
679 hD DATA 00,00,ff,ff,00,00,ff,ff,00,00
680 Kh DATA ff,ff,00,00,7f,ff,0c,00,7f,ff
681 uJ DATA 00,00,3f,ff,00,00,0f,ff,00,00
682 Re DATA 07,ff,00,00,70,fe,a0,62,b0,00
683 BD DATA 00,00,00,00,00,00,00,00,00,00
684 kP DATA e0,00,00,00,f8,00,00,04,f8,00
685 cO DATA 04,00,f8,00,06,00,f8,00,07,fe
686 dF DATA f8,00,07,fe,ff,fe,07,fe,f8,00
687 5v DATA 06,00,f8,00,04,00,f8,00,04,00
688 UK DATA f8,00,00,00,f8,00,00,00,00,00
689 af DATA 00,0c,c0,00,00,00,00,00,00,00
690 ix DATA 00,00,01,20,00,00,01,22,00,00
691 wM DATA 01,24,00,00,01,26,00,00,01,28
692 UD DATA 00,00,01,2a,00,00,01,2c,00,00
693 58 DATA 01,2e,00,00,01,30,00,00,01,32
694 JX DATA 00,00,01,34,00,00,01,36,00,00
695 A1 DATA 01,38,00,00,01,3a,00,00,01,3e
696 re DATA 00,00,01,3e,00,00,0c,e0,00,00
697 WN DATA 00,e2,00,00,00,e4,0c,00,00,e6
698 Nz DATA 00,00,00,e8,00,00,00,e2,00,00
699 ZE DATA 00,e0,00,00,00,ea,00,00,ff,ff
700 uA DATA ff,fe,7c,01,eb,6f,0e,00,66,c3
701 nW DATA 36,6c,33,00,66,c3,33,ee,33,00
702 MI DATA 6c,c3,f1,8f,36,00,60,c3,31,8c
703 g6 DATA 36,00,60,c3,31,8e,33,00,60,fb
704 Sh DATA 31,8f,33,00,38,18,18,18,18,18
705 uN DATA 18,00,3e,66,06,1c,30,60,7e,00
706 wa DATA 3e,6c,06,1e,06,66,3e,00,66,66
707 7K DATA 66,3e,06,06,06,00,7e,6c,7e,06
708 Be DATA 06,06,7c,0c,3e,60,60,7e,66,66
709 GJ DATA 3e,00,7c,0c,0c,18,18,3c,30,00
710 zF DATA 3e,66,66,3e,66,66,3e,00,3e,66
711 lI DATA 66,3e,06,06,3e,00,3e,66,66,66
712 hW DATA 66,66,3e,00,00,31,5a,75,7f,75
713 JS DATA 5a,31,00,ef,a6,8b,81,8b,a6,ef
714 zy DATA 70,80,3a,b7,64,a0,80,80,80,9f
715 lI DATA 5c,08,60,80,80,b1,3f,b1,10,88
716 mN DATA 60,80,9a,3f,7f,84,80,4a,f3,ff
717 OJ DATA 3b,42,d3,80,6c,3f,74,24,c0,80
718 Os DATA 80,97,4b,68,f0,84,80,1f,34,d0
719 J2 DATA e4,80,bf,bf,d7,00,2f,eb,80,80
720 Za DATA c1,3f,20,ab,8c,60,82,f1,c0,eb
721 7K DATA 80,80,c0,20,00,f7,e0,a0,80,9f
722 Vc DATA 30,02,c3,ba,80,9b,dF,f8,c0,a9
723 4z DATA 99,a6,37,19,1e,a4,60,9c,ef,cc
724 db DATA f3,e0,b0,94,8b,88,04,d0,bf,aa
725 XI DATA a8,a1,0f,j8,ca,87,90,b7,e,f2

```

Listing 1. (Fortsetzung)

726 78 DATA f8,a2,f0,a7,f2,f3,cc,a8,92,a7
727 78 DATA 08,f1,c8,b7,8c,a2,f7,07,e0,d9
728 78 DATA e9,b5,q3,bc,86,d7,ca,d1,b0,q4
729 78 DATA d3,e1,d1,f7,9a,c9,cf,62,ee,d0
730 78 DATA 8c,b7,d0,f7,f1,8f,9e,97,ce,be
731 78 DATA df,1c,a7,c5,d0,e5,xc,cc,d7,3c
732 78 DATA ec,80,b3,ff,b8,b7,e5,c2,88,9b
733 78 DATA 17,28,d8,80,97,bd,f7,f3,a8,90
734 78 DATA 99,c9,f8,f4,b5,97,9f,ad,dt,eb
735 78 DATA e0,d1,8f,c4,f5,f4,ba,ae,b4,d9
736 78 DATA a7,ca,e7,e9,8f,ef,a6,e4,00
737 78 DATA e2,89,95,bf,dt,de,16,b3,84,a7
738 78 DATA f1,f3,d2,xc,8c,a7,c3,d7,00,b8
739 78 DATA 9d,b0,bf,bf,e0,e5,ac,a2,bf,e7
740 78 DATA ea,d9,ed,bc,e7,c4,d7,ce,18,b3
741 78 DATA e7,d0,e7,dc,b3,af,b0,17,11,b8
742 78 DATA ct,cc,e0,93,bf,e0,df,c8,b2,bd
743 78 DATA e0,d7,e9,d4,e4,c7,c8,cf,d2,ce
744 78 DATA c9,b6,ff,d4,ea,d5,bf,c4,e2,17
745 78 DATA dc,cf,c9,2f,c7,cc,b0,d1,d0,c1
746 78 DATA b3,d0,ab,e7,ef,cd,b6,b3,b9,c8
747 78 DATA c9,c0,b1,b3,c4,c4,c3,c4,ce,ad
748 78 DATA c0,cf,d3,eb,6c,c0,bf,17,11,ef
749 78 DATA d3,co,c3,b7,d0,d7,c0,ba,b3,ce
750 78 DATA d8,d7,ec,d1,c6,88,b3,db,d4,c6
751 78 DATA b4,b8,b5,ce,d7,c3,b0,af,b0,e0
752 78 DATA d6,cc,b3,b4,c4,c2,ce,c3,c5
753 78 DATA c0,c5,d1,b3,13,c2,ce,bf,e3,df
754 78 DATA d0,c0,af,bf,d8,d1,c3,bf,b7,b0
755 78 DATA c7,17,d0,c0,b7,88,eb,d0,de,d3
756 78 DATA b8,be,c5,d0,d3,bf,e0,bf,bf,c8
757 78 DATA cd,c3,cc,c1,cb,d3,cd,d0,ef,c9
758 78 DATA cc,bf,ba,ce,cf,c8,ed,b7,c4,d7
759 78 DATA d3,d0,bd,b5,c6,c4,ce,ef,b9,b0
760 78 DATA cf,c9,c8,cc,c3,cc,c3,c2,ca,c5
761 78 DATA c2,c7,c5,c4,cd,c7,88,c2,3c,d6
762 78 DATA cd,ca,cb,cd,d6,d3,c4,ca,d3,cb
763 78 DATA ce,c9,ca,ca,bd,c8,d3,c5,c4,c5
764 78 DATA c0,c4,bf,c1,c4,bf,c0,c7,c6,c8
765 78 DATA c3,c3,c6,c7,c5,c8,cf,1c,cf,c4
766 78 DATA c0,d3,ca,c4,bd,cb,d0,bf,c2,ef
767 78 DATA c7,c4,c7,c1,c4,c9,c4,c8,c7,c2
768 78 DATA cf,cf,d0,c7,cb,c8,cb,c5,c4,bf
769 78 DATA a3,c8,c7,c6,c7,cb,c9,cf,d1,d0
770 78 DATA cf,d5,d8,d2,cc,cd,c7,c8,ef,c7
771 78 DATA c8,c4,b0,c8,c9,c0,ba,b3,b0,b7
772 78 DATA bf,c8,bf,bd,c8,c9,cf,d1,d8,ca,c0
773 78 DATA c7,cd,d0,cf,c5,c6,c7,c9,d4,d5
774 78 DATA dc,cf,c7,c8,cb,c7,ca,c9,b0,bc
775 78 DATA be,06,ce,c9,c9,c3,c2,c4,c7,c2
776 78 DATA c4,c3,ba,bf,c3,c9,cf,c7,c1,cb
777 78 DATA cf,d2,cf,c3,c6,c5,c9,cf,ce,cc
778 78 DATA cb,c9,cc,cf,c7,cc,cb,c2,c6,c9
779 78 DATA c9,cf,c4,cc,16,c5,cc,d3,ce,d2
780 78 DATA cb,cd,cd,cb,c8,cd,c5,c4,c7,c9
781 78 DATA d0,c7,c3,ca,c7,cc,d3,ce,cc,c6
782 78 DATA c3,ce,cd,cc,cc,bf,ca,cd,cb,c8
783 78 DATA c5,c1,cc,cb,cc,18,ca,cb,cf,c8
784 78 DATA c4,c7,c6,cc,cc,c6,ca,13,cc,cb
785 78 DATA c9,c4,c7,c3,c0,c3,c1,c4,bd,bf
786 78 DATA a6,c7,c4,c7,c9,c8,cb,c7,d0,cf
787 78 DATA c9,d8,d3,d7,cf,cf,ca,bc,c0,d0
788 78 DATA cd,c9,d0,ca,c0,c4,c2,c0,c3,bf
789 78 DATA c2,c7,c3,c8,c7,c3,cc,c7,c8,cd
790 78 DATA cb,d2,13,cf,d0,cf,cc,14,1c,d0
791 78 DATA d5,cf,00,d3,c5,c8,c7,c5,ce,c9
792 78 DATA c0,c7,c2,cc,cc,bd,cc,bf,be,c8
793 78 DATA c3,c3,ca,cc,cc,18,ca,cb,cf,c8
794 78 DATA cc,cf,c4,cc,cc,cc,c4,cc,1c,c9
795 78 DATA c7,80,cf,cd,b0,ca,c4,cd,c7,c8
796 78 DATA c7,15,cb,cc,cd,c9,c3,cc,1c,cd
797 78 DATA ca,11,c9,cc,cd,cd,cd,1c,1c,ce
798 78 DATA cb,c9,ce,c9,c8,cb,c7,cc,cd,cb
799 78 DATA d0,cd,c8,cf,c7,c4,c7,c1,c2,c2
800 78 DATA c1,c2,bd,d9,cc,cf,c3,cf,c9,ce
801 78 DATA cd,c7,dc,ca,cc,cc,cc,13,d1
802 78 DATA df,cf,cb,1c,13,c9,cc,cb,c4,c7
803 78 DATA c3,xc,c7,1c,cf,c9,c3,cc,1c,c5
804 78 DATA ca,c3,c8,c7,bf,c4,c7,c6,ca,cb
805 78 DATA c9,cc,c9,c9,cc,ca,1c,1c,cc,cc
806 78 DATA cf,cd,dc,cb,ca,d3,c7,ca,d5,d3
807 78 DATA d1,ce,cf,c9,c7,c5,ca,bf,1c,d0,cf
808 78 DATA bd,c0,c3,c0,c8,c5,c0,c8,c3,c3

809 T3 DATA cb,c9,cc,cf,cb,d0,d2,cd,d4,ce
810 0g DATA ce,d5,cf,d2,d3,cf,d0,cf,c7,c8
811 2S DATA c6,c1,ca,c3,c4,c7,c1,c0,c5,c6
812 4e DATA ce,c9,c1,c8,c3,c5,ce,c7,c8,cb
813 xc DATA c5,c8,c4,c4,c8,c5,c3,ce,cb,ce
814 9n DATA d3,d1,d2,d7,d4,dc,d9,d5,dc,d3
815 J. DATA d2,d7,d3,d8,86,d2,d2,cf,ce,ce
816 xV DATA c6,c0,c7,bd,c0,bf,b7,c0,bb,b9
817 nJ DATA c4,bf,c0,c6,be,c0,bf,be,c0,c3
818 6V DATA be,c8,c9,c8,ce,c7,ca,cf,c9,c9
819 Jn DATA cd,c7,cc,cb,c8,d2,cf,d0,d7,d1
820 Y4 DATA cc,cf,c9,dc,cd,c5,cc,c7,c8,cc
821 7A DATA c7,c4,cb,c7,c8,cb,c7,cc,cc,c9
822 r DATA ce,c4,c1,c7,c3,17,00,00,00,00
823 X DATA 00,00,00,00,00,00,00,00,00,00
824 Y DATA 00,00,00,00,00,00,00,00,00,00
825 J2 DATA 00,00,00,00,00,00,00,00,00,00
826 ka DATA 00,00,00,00,00,00,00,00,00,00
827 Jb DATA 00,00,00,00,00,00,00,00,00,00
828 Me DATA 00,00,00,00,00,00,00,00,00,00
829 Nd DATA 00,00,00,00,00,00,00,00,00,00
830 Pe DATA 00,00,00,00,00,00,00,00,00,00
831 Pf DATA 00,00,00,00,00,00,00,00,00,00
832 Qg DATA 00,00,00,00,00,00,00,00,00,00
833 rI DATA 00,00,03,ee,00,00,01,53,00,00
834 rI DATA 00,00,00,00,18,8e,00,00,18,80
835 Bp DATA 00,00,12,72,00,00,18,64,00,00
836 cx DATA 18,56,00,00,18,48,00,00,18,3a
837 En DATA 00,00,18,2a,00,00,18,1e,00,00
838 b7 DATA 18,10,00,00,17,36,00,00,17,2c
839 SB DATA 00,00,17,24,00,00,17,1c,00,00
840 gH DATA 17,12,00,00,17,08,00,00,16,f0
841 qH DATA 00,00,16,40,00,00,16,c8,00,00
842 09 DATA 16,c2,00,00,16,b6,00,00,16,ae
843 A DATA 00,00,16,40,00,00,16,9a,00,00
844 30 DATA 16,94,00,00,16,8c,00,00,16,84
845 eC DATA 00,00,16,7c,00,00,16,74,00,00
846 b3 DATA 16,6a,00,00,16,62,00,00,16,5a
847 17 DATA 00,00,16,52,00,00,16,48,00,00
848 nV DATA 16,40,00,00,16,36,00,00,16,2c
849 SE DATA 00,00,16,24,00,00,16,1c,00,00
850 0n DATA 15,ea,00,00,15,e0,00,00,15,ea
851 G3 DATA 00,00,15,b0,00,00,15,7e,00,00
852 Jz DATA 15,34,00,00,14,fc,00,00,14,e2
853 ar DATA 00,00,14,d2,00,00,14,e2,00,00
854 Qm DATA 14,b6,00,00,14,e6,00,00,14,a0
855 p7 DATA 00,00,14,9a,00,00,14,8e,00,00
856 89 DATA 14,96,00,00,14,7a,00,00,14,6e
857 2m DATA 00,00,14,66,00,00,14,5e,00,00
858 1p DATA 14,58,00,00,14,4a,00,00,14,34
859 1y DATA 00,00,14,2e,00,00,14,26,00,00
860 Pf DATA 14,20,00,00,14,12,00,00,13,fa
861 K0 DATA 00,00,13,f2,00,00,13,e6,00,00
862 Kv DATA 13,d8,00,00,13,d8,00,00,13,d0
863 Jz DATA 00,00,13,c4,00,00,13,bc,00,00
864 J9 DATA 13,b4,00,00,13,ac,00,00,13,a6
865 Jf DATA 00,00,13,9a,00,00,13,92,00,00
866 Q DATA 13,8a,00,00,13,7e,00,00,13,68
867 nW DATA 00,00,13,62,00,00,13,52,00,00
868 eA DATA 13,4a,00,00,13,42,00,00,13,36
869 11 DATA 00,00,13,2e,00,00,13,26,00,00
870 01 DATA 13,04,00,00,12,fa,00,00,12,f4
871 ce DATA 00,00,12,ec,00,00,12,e4,00,00
872 0v DATA 12,da,00,00,12,a6,00,00,12,8a
873 dg DATA 00,00,12,86,00,00,12,7a,00,00
874 vp DATA 12,70,00,00,12,66,00,00,12,5e
875 xq DATA 00,00,12,58,00,00,12,4e,00,00
876 pJ DATA 12,46,00,00,12,40,00,00,12,3a
877 11A DATA 00,00,12,34,00,00,12,2a,00,00
878 56 DATA 12,22,00,00,12,1c,00,00,12,16
879 7L DATA 00,00,12,10,00,00,12,06,00,00
880 fT DATA 11,fa,00,00,11,ea,00,00,11,e6
881 Xf DATA 00,00,11,e0,00,00,11,d6,00,00
882 Me DATA 11,ca,00,00,11,c8,00,00,11,be
883 yb DATA 00,00,11,b2,00,00,11,aa,00,00
884 27 DATA 11,a4,00,00,11,9e,00,00,11,98
885 2J DATA 00,00,11,8a,00,00,11,82,00,00
886 0N DATA 11,76,00,00,11,6e,00,00,11,6a
887 7x DATA 00,00,11,5a,00,00,11,44,00,00
888 x6 DATA 11,3e,00,00,11,34,00,00,11,26
889 V DATA 00,00,11,1a,00,00,11,12,00,00
890 n. DATA 11,00,00,00,10,e8,00,00,10,d8
891 1m DATA 00,00,10,ce,00,00,10,b8,00,00

892 d0 DATA 10,a2,00,00,10,96,00,00,10,8e
893 67 DATA 00,00,10,82,00,00,10,76,00,00
894 J6 DATA 10,6a,00,00,10,5e,00,00,10,52
895 1k DATA 00,00,10,46,00,00,10,3a,00,00
896 nb DATA 10,2e,00,00,10,1a,00,00,10,08
897 xx DATA 00,00,0f,ec,00,00,0f,e4,00,00
898 f1 DATA 0f,d2,00,00,0f,c0,00,00,0f,ae
899 ad DATA 00,00,0f,92,00,00,0f,8a,00,00
900 1V DATA 0f,78,00,00,0f,66,00,00,0f,54
901 wZ DATA 00,00,0f,38,00,00,0f,30,00,00
902 0X DATA 0f,1e,00,00,0f,0c,00,00,0f,f0
903 Fz DATA 00,00,0e,88,00,00,0e,d6,00,00
904 1r DATA 0e,c4,00,00,0e,a8,00,00,0e,a0
905 1p DATA 00,00,0e,96,00,00,0e,7a,00,00
906 1i DATA 0e,64,00,00,0e,54,00,00,0e,36
907 1r DATA 00,00,0e,18,00,00,0e,fa,00,00
908 1A DATA 0d,e0,00,00,0d,d8,00,00,0d,d0
909 14 DATA 00,00,0d,c6,00,00,0d,bc,00,00
910 v4 DATA 0d,a6,00,00,0d,ad,00,00,0d,92
911 16 DATA 00,00,0d,86,00,00,0d,7a,00,00
912 1b DATA 0d,6c,00,00,0d,54,00,00,0d,44
913 1n DATA 00,00,0d,38,00,00,0d,24,00,00
914 1n DATA 0d,0e,00,00,0d,02,00,00,0d,fa
915 1d DATA 00,00,0d,ea,00,00,0d,e2,00,00
916 1x DATA 0c,d6,00,00,0c,0a,00,00,0c,be
917 5d DATA 00,00,0c,b2,00,00,0c,aa,00,00
918 1c DATA 0c,98,00,00,0c,86,00,00,0c,74
919 21 DATA 00,00,0c,58,00,00,0c,50,00,00
920 21 DATA 0c,3a,00,00,0c,2e,00,00,0c,1a
921 22 DATA 00,00,0b,fa,00,00,0b,f6,00,00
922 1f DATA 0b,e4,00,00,0b,d2,00,00,0b,c0
923 10 DATA 00,00,0b,a4,00,00,0b,9a,00,00
924 7c DATA 0b,8a,00,00,0b,78,00,00,0b,5c
925 1b DATA 00,00,0b,54,00,00,0b,42,00,00
926 ad DATA 0b,30,00,00,0b,14,00,00,0b,0c
927 vp DATA 00,00,0b,02,00,00,0b,ea,00,00
928 29 DATA 0a,d0,00,00,0a,c0,00,00,0a,a2
929 6b DATA 00,00,0a,84,00,00,0a,68,00,00
930 7c DATA 0a,52,00,00,0a,4a,00,00,0a,3c
931 J1 DATA 00,00,0a,34,00,00,0a,2c,00,00
932 1f DATA 0a,22,00,00,0a,18,00,00,0a,12
933 71 DATA 00,00,0a,04,00,00,0a,f0,00,00
934 3a DATA 09,f2,00,00,09,86,00,00,09,d2
935 ep DATA 00,00,09,ca,00,00,09,a2,00,00
936 7c DATA 09,ba,00,00,09,ae,00,00,09,a6
937 2r DATA 00,00,09,8e,00,00,09,82,00,00
938 32 DATA 09,7a,00,00,09,66,00,00,09,36
939 1c DATA 00,00,09,2e,00,00,09,1a,00,00
940 12 DATA 09,16,00,00,09,06,00,00,09,fa
941 1f DATA 00,00,08,f2,00,00,08,eb,00,00
942 pN DATA 08,da,00,00,08,d2,00,00,08,aa
943 1w DATA 00,00,08,bc,00,00,08,b2,00,00
944 1c DATA 08,a6,00,00,08,9a,00,00,08,96
945 11 DATA 00,00,08,8a,00,00,08,82,00,00
946 1q DATA 08,7a,00,00,08,6a,00,00,08,62
947 1c DATA 00,00,08,58,00,00,08,4a,00,00
948 XF DATA 08,40,00,00,08,36,00,00,08,2c
949 27 DATA 00,00,08,18,00,00,08,12,00,00
950 1c DATA 07,fa,00,00,07,f2,00,00,07,ea
951 1c DATA 00,00,07,e2,00,00,07,d8,00,00
952 K5 DATA 07,c2,00,00,07,ac,00,00,07,a4
953 18 DATA 00,00,07,92,00,00,07,7a,00,00
954 1K DATA 07,70,00,00,07,64,00,00,07,38
955 16 DATA 00,00,07,18,00,00,07,12,00,00
956 cf DATA 07,08,00,00,07,e4,00,00,07,be
957 17 DATA 00,00,06,ae,00,00,06,74,00,00
958 1c DATA 06,54,00,00,06,4e,00,00,06,44
959 1a DATA 00,00,06,20,00,00,06,fa,00,00
960 1e DATA 05,ea,00,00,05,44,00,00,05,1a
961 1v DATA 00,00,05,12,00,00,05,04,00,00
962 2c DATA 04,f0,00,00,04,e8,00,00,04,a0
963 K3 DATA 00,00,04,84,00,00,04,c8,00,00
964 1c DATA 04,ae,00,00,04,94,00,00,04,7a
965 10 DATA 00,00,04,d2,00,00,04,cc,00,00
966 20 DATA 00,64,00,00,04,5e,00,00,04,58
967 d1 DATA 00,00,04,52,00,00,04,4c,00,00
968 1d DATA 00,46,00,00,04,40,00,00,04,3a
969 1c DATA 00,00,04,34,00,00,04,1a,00,00
970 zw DATA 00,00,00,00,03,f2
(C) 1989 M&T

Listing 1. (Schluß)

Krieg der Sterne

Es war einmal in ferner Zukunft... So wie der Film »Krieg der Sterne« könnte auch »Fantasy« beginnen; ein tolles Spiel, das in einer weit vor uns liegenden Zeit abläuft.

von Holger Hoffmann

Wir schreiben das Jahr 6533. In einer Welt aus 59 Planeten kämpfen einige Planeten um die Vorherrschaft. Während auf den Heimatplaneten die Produktion auf Hochtouren läuft, beginnen die Herrscher andere Planeten zu erobern. Lassen Sie sich von »Fantasy« in diese Welt entführen. Aber nur, wenn Sie genügend Zeit haben, denn Fantasy wird Sie lange fesseln.

Nach dem Abtippen (siehe Kasten mit Hinweisen), kann das Abenteuer für ein bis vier Spieler beginnen. Fantasy wird, bis auf die Eingabe von Zahlen und Namen, komplett mit der Maus gesteuert. Am rechten Bildschirm finden Sie jeweils die Menüpunkte in Form von Schaltern.

■ **Spiel laden:** Hiermit laden Sie ein vorher gespeichertes Spiel von Diskette. Der Dateiname wird mit Pfad (z.B. Df1:Welten/Welt_2) eingegeben. Anschließend beginnt das Spiel.

■ **Neues starten:** Fantasy kann mehrere Welten verwalten. Für jede Welt besteht die Möglichkeit, mehrere Spiele zu speichern. Mit diesem Menüpunkt starten Sie ein neues Spiel auf einer schon generierten Welt (siehe auch »Welt kreieren«). Nach dem Anwählen erscheint ein Fenster, in dem Sie nun den Namen der gewünschten Welt eingeben. Danach müssen Sie noch die Fragen nach der Zahl und den Namen der Spieler beantworten.

■ **Welt kreieren:** Sie können mehrere verschiedene Welten anlegen. Nach Anwählung dieses Schalters zeichnet Fantasy dreimal ein Dreieck in die linke obere Ecke des Bildschirms. Nach diesem Vorgang geben Sie den Namen zur Speicherung auf Diskette ein.

■ **PRG - beenden:** Dient zum Verlassen des Spiels und des Basic-Interpreters. Wenn ein Spiel geladen oder ein neues gestartet wurde, erscheint das eigentliche Spielfeld.

■ **Bauen:** Es erscheint ein Untermenü mit neuen Schaltern:
- **Kriegsschiffe:** Pro Schiff müssen auf dem Heimatplaneten (in der Grafik von einem Kreis umgeben) drei Geldeinheiten sowie vier Rohstoffeinheiten vorhanden sein. Sie erhalten die neuen Schiffe in der nächsten Runde.

- **Transportschiffe:** Zum Bau benötigt man pro Schiff eine Geldeinheit und drei Rohstoffeinheiten.

- **Gewerbezentrum:** Um von einem Planeten Rohstoffe abtransportieren zu können, muß man bestimmte Lizenzen zahlen und Verladeanlagen einrichten. Das kostet 50 Geldeinheiten.

- **Industriezentrum:** Für die »lächerliche« Summe von 500 Geldeinheiten können Sie Ihren Heimatplaneten (Produktion) verlegen.

- **Weiter:** Dient zur Rückkehr ins Hauptmenü.

■ **Angriff:** Hier zeigt sich die komfortable Seite von Fantasy. Nach

Holger Hoffmann

Anfang 1984 stieg Holger Hoffmann in den C64-Boom mit ein. Nach wenigen Wochen erfolgte der Umstieg von Basic auf Assembler. Nach dem Preissturz des Amiga legte sich der Autor einen Amiga 500 zu. Auch hier erfolgte der Start in Basic. Neben Assembler arbeitet Holger Hoffmann noch mit C und Modula-2. Sein Studienwunsch ist Mathematik-Informatik, sein nächster Hardware-Wunsch eine Festplatte.

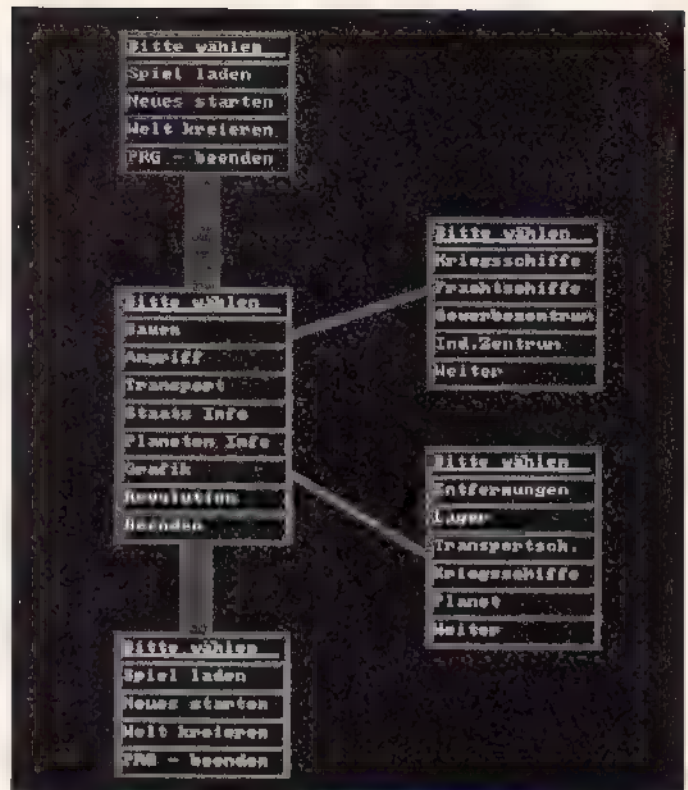
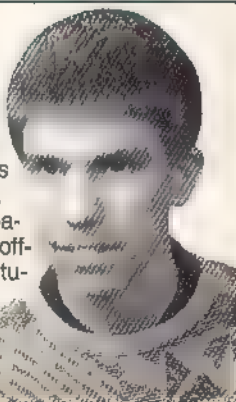


Bild 1. Die Menüstruktur von »Fantasy«. Drei Untermenüs gibt es. Das unten stehende Menü erreichen Sie über »Beenden«.

dem Anklicken des Schalters wählen Sie den Planeten, von dem Sie Schiffe losschicken wollen, per Mausklick an. Danach bestimmen Sie den Zielplaneten ebenfalls durch Anklicken. Nun verlangt Fantasy die Eingabe, wie viele Schiffe starten sollen. Bevor Sie versuchen einen Planeten zu erobern, sollten Sie ein einzelnes Schiff losschicken. Ist keine Bevölkerung vorhanden, besetzen Sie den Planeten sofort. Andernfalls baut die Besatzung einen Spionagering auf. Ob der Angriff erfolgreich war, zeigt sich in der nächsten Spielrunde.

■ **Transport:** Genau wie bei den Angriffen, legen Sie den Start- und Zielplaneten durch Mausklick fest. Zusätzlich zur Anzahl der zu verwendenden Schiffe, geben Sie noch die Zahl der zu transportierenden Rohstoffe ein. Pro Schiff ist eine Einheit möglich. Wichtig ist, daß beide Planeten Ihnen gehören müssen.

■ **Staats Info:** Das Programm zeigt wichtige Daten über Ihr Imperium an. Bei der Zahl der Schiffe ist zu beachten, daß Fantasy die in diesem Spielzug schon bewegten Schiffe nicht berücksichtigt.

■ **Planeten Info:** Nach Anklicken eines Planeten erhalten Sie Informationen.

■ **Grafik:** Natürlich kann sich niemand merken, wieviel Schiffe und Rohstoffe wo zu finden sind. Darum zeigt Ihnen Fantasy solche Informationen grafisch an. Im Untermenü gibt es folgende Punkte:

- **Entfernungen:** Durch mehrmaliges Anklicken des Schalters wählen Sie die drei Einstellungen.

- **Lager:** Alle Planeten mit über 30 Einheiten an Rohstoff werden blau eingefärbt. Bei mehr als 150 Einheiten wird er grau dargestellt.

- **Transportsch.:** Blau bedeutet mehr als 10, grau mehr als 50 Transportschiffe auf dem Planeten. Ansonsten ist der Planet braun.

- **Kriegsschiffe:** Hier gelten dieselben Angaben wie bei den Transportschiffen.

- **Planet:** Bei hohem Steuereinkommen ist der Planet blau, bei hohem Rohstoffvorkommen grau.

- **Weiter:** Rückkehr ins Hauptmenü.

■ **Revolution:** Hin und wieder finden auf den Planeten Revolutionen statt. Um auf einem fremden Planeten eine Revolution zu begünstigen, können Sie Geld investieren. Klicken Sie dazu einfach auf den entsprechenden Planeten. Gelingt die Revolution, gehört der Planet anschließend Ihnen. Damit ein Aufstand überhaupt möglich ist, muß sich ein Spionagering auf dem Planeten aufhalten (siehe auch »Angriff«). Pro Runde findet nur ein Aufstand statt. Wählen mehrere Spieler diesen Punkt an, so findet der zweite eine Runde später statt. Noch eine schlechte Nachricht: Auch auf den

von Ihnen besetzten Planeten kann ein Aufstand stattfinden. Ist er erfolgreich, verlieren Sie den Planeten.

■ **Beenden:** Im Mehrspielermodus übergeben Sie hiermit die Kontrolle an den nächsten Herrscher. Damit alle anderen Mitspieler wegsehen können, erscheint anschließend ein Fenster, in dem der nächste Herrscher genannt wird. Nach dem letzten Spieler erscheint ein Menü, in dem man laden und speichern kann. Bei nur einem Spieler erscheint dieses Menü sofort.

■ **Save Game:** Speichert den aktuellen Spielstand.

■ **Load Game:** Lädt ein vorher gespeichertes Spiel.

■ **Ende:** Führt zu dem am Anfang erklärten Menü zurück.

■ **Weiter:** Die nächste Spielrunde wird gestartet.

Das Spiel endet, wenn nur noch ein Herrscher seinen Heimatplaneten besitzt. Der Einspielermodus dient nur zu Trainingszwecken und endet nicht. Aber auch alleine macht Fantasy viel Spaß und außerdem lernt man so Tricks und Kniffe, damit man beim nächsten Turnier der Stärkere ist.

kn

Hinweise zum Abtippen

Damit Sie Fantasy spielen können, müssen einige Vorbedingungen erfüllt sein:

- Folgende Zeichensätze müssen vorhanden sein:

diamond 20 opal 9 sapphire 14

Dies ist auf alle Fälle gewährleistet, wenn Sie von Ihrer Original-Workbench-Diskette starten.

- Die folgenden Dateien müssen im selben Verzeichnis stehen wie Amiga-Basic und das Programm "Fantasy":

intuition.bmap graphics.bmap diskfont.bmap

Fantasy benötigt viel Speicher. Bei Amigas mit 512 KByte RAM kann das zu Problemen führen. Diese lassen sich jedoch umgehen. Schalten Sie den Computer aus, damit erreicht man, daß nach dem Einschalten der gesamte Speicher zur Verfügung steht. Schließen Sie alle CLIFenster. Starten Sie das Programm von der Workbench aus durch Doppelklick auf das Programmsymbol. Starten Sie kein anderes Programm. Auf keinen Fall dürfen Sie das CLI-Kommando ADDBUFFERS einsetzen.

Programmname: Fantasy

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: Basic

Bemerkung: siehe Kasten

Programmautor: Holger Hoffmann

```
1 NxC CLEAR,52000h
2 BI DECLARE FUNCTION opendiskfont& LIBRARY
3 MA DECLARE FUNCTION opendiskfont LIBRARY
4 qf DECLARE FUNCTION opendiskfont LIBRARY
5 th DECLARE FUNCTION opendiskfont LIBRARY
6 HU LIBRARY "intuition.library"
7 l7 LIBRARY "graphics.library"
8 pz LIBRARY "diskfont.library"
9 vF GOSUB font:GOSUB titel:RANDOMIZE TIMER:GOSUB sar
10 AI FOR x= 0 TO 7:PALETTE x,0,0,0:NEXT
11 mo DIM ex(826),g1(4),g2(4),s1(4),s2(4)
12 oV DIM kx(59),ky(59),k1(4,50),k2(4,50),k3(4,50),del(4,50),hel(4,50),de2(4,50)
13 Pt DIM ro(59),mo(59),la(59),sh(59),tr(59)
14 tq DIM na(59),st(4,59),na3(4),f2(4),f3(4),f4(4),f5(4),v(4)
15 KL DIM a1(20),a2(20),a3(20),a4(20),re(59)
16 rU DIM st3(5),va(600),ha(3),heia(4),f1(4),gf3(3200)
17 lh st3(0)="Existenz bekannt":st3(1)="bekannt"
18 oQ st3(2)="besitzend":st3(3)="gewerblich nutzend"
19 AX st3(4)="industriell nutzend"
20 BY sr(1)=31:sr(2)=33:sr(3)=38
21 Gs f2(0)=2:f3(0)=1:GOSUB planeter:CLS
22 JZ na3(0)="Ureinwohner"
23 Uw neu=PALETTE 0,0,0,0
24 qd PALETTE 1,1,1,1:PALETTE 2,0,0,0:PALETTE 3,5,5,1
25 WN PALETTE 4,0,0,1:PALETTE 5,7,3,0
26 la PALETTE 6,5,5,5:PALETTE 7,1,0,0
27 qJ ON ERROR GOTO error
28 y2 start:
```

```
29 y1 CLS:GOSUB menuel
30 xW ***
31 a1 GOSUB eingabe
32 X3 IF wa=1 THEN GOSUB lod ar s3<>" THEN anfang:ELSE GOTO w
w5
33 er IF wa=2 THEN GOTO neu
34 lx IF wa=3 THEN GOSUB grafik:GOTO start
35 yw IF wa=4 THEN MENU RESET:GOSUB cfont:WINDOW CLOSE 1:SCREEN C
LOSE 1:END
36 d1 hTC wa5
37 zC ***
38 AX GOSUB wldc IF s3=" THEN start
39 lA GOSUB zufall:GOSUB meiviele
40 rv fahr=1
41 CP anfang
42 BI FOR i= 1 TO spieler
43 G12 IF a(heia(1))<>1 THEN wa3
44 AX g1=(g2+s0=s1/s2=s0 s2=s0
45 Ww na3="Runde :"+s0+s1+s2+s3
46 G10 wv4
47 Ca2 IF LEV(na3)<12 THEN na3=na3+" ":GOTO wv4
48 rK re3=na3+" "
49 OY CALL SetWindowTitle(WINDOW(7),SADD(na3),0)
50 Tu GOSUB erf
51 MA CLS:GOSUB menu2
52 WP IF spieler<>1 THEN GOSUB nex
53 W8 di=f1(1):GOSUB dls
54 G10 wv2
55 E12 GOSUB eingabe
56 pl IF m= 8 THEN GOTO wv3
57 We IF m=4 THEN GOSUB einkommen
58 dV IF m=1 THEN GOSUB bauen
59 b6 IF m=6 THEN GOSUB graf
60 Ny IF (m<>0) AND (wa<>0) THEN ON wa GOSUB bauen,senden,t
ransl,info,info,info,info,info,info
61 yv GOTO wv2
62 Ls0 wv3:
63 -g2 p=heia sr,p=shp+g1 tr p=tr pl+g2(1)
64 Zs0 NEXT i
65 JC sr=ar+1
66 HQ COLOR 1,0:CLS:FOR p= 1 TO 59:la=ra p
67 u22 IF la<>0 AND st(la,p)>1 THEN v=la+v+mo p la,p)=
a(p)+ro(p)
68 BGO NEXT
69 bR na3="Fantasy Gewinner in 1986 by Holger Hoffmann"+CHR$(0)
70 jt CALL SetWindowTitle(WINDOW(7),SADD(na3),0)
71 Uc FOR x= 1 TO spieler
72 H12 IF s1(1)=0 THEN wv7
73 On sr=1 TO s1(1)-1 GOSUB war NEXT w
74 v2 s1(1)=0
75 oP0 wv7.
76 WD2 IF s2(1)=0 THEN wv8
77 zc FOR x=0 TO s2(1)-1
78 s34 pa=de2(1)+x pa p=na p+v3 l,x
79 jd tr p=tr p-k2(1)+x
80 F12 NEXT w
81 G10 s2(1)=0
82 ab wv8
83 oB wv7
84 MZ2 pa=INT(ha(1)+x+1
85 AZ FOR x= 1 TO 59:IF na(x)=0 THEN sh(x)=sh(x)+INT((mo(x)+ro(
x))/10)
86 Er4 IF re(x)>re(pl) THEN pl=x
87 Kp2 NEXT x
88 t9 IF re(pl)=0 THEN zu=RND(1) ELSE zu=1
89 -p h=0:FOR x=1 TO spieler:IF pl=heia(x) THEN h=1
90 Na NEXT x
91 tN GOSUB menu2
92 G12 IF (zu>.4) THEN GOSUB rev
93 zc0 wv6
94 rv2 GOSUB eingabe
95 dV IF wa=1 THEN GOSUB sav
96 bj IF wa=2 THEN GOSUB lod
97 st IF wa=3 THEN GOSUB bauen
98 la IF wa<>4 THEN wv6
99 lr en=0
100 T4 FOR x= 1 TO spieler IF na=heia x, -x THEN en=en+1
101 Y3 NEXT x
102 d3 ar spi er=1 (5 er>1) THEN GOTO anfang
103 48 SCREEN CLOSE 1:WINDOW CLOSE 1
104 kV GOTO sieg
105 Xf0 dars:
```



```

106 'u d1=d1+1 : d1=4 THEN GOTO 107
107 g6 GOSUB dis:RETURN
108 Lr planeten:
109 b0 DATA 0,0,0,0,0,0,5,5,5,5,0,0,0,0,0,0
110 p8 DATA 0,0,0,5,7,5,5,5,5,0,5,0,5,0,0,0
111 gF DATA 0,5,5,7,5,5,7,5,5,5,0,5,5,0,5,0
112 jD DATA 5,5,7,5,5,7,5,5,0,5,5,0,5,5,0,5
113 e2 DATA 5,7,5,5,7,5,5,7,5,5,0,5,0,5,0,5
114 aT DATA 0,5,7,5,5,5,7,5,5,5,5,5,5,0,5,0
115 Py DATA 0,0,0,5,5,7,5,7,5,5,5,0,5,0,0,0
116 qX DATA 0,0,0,0,0,0,7,5,5,5,0,0,0,0,0,0
117 R7 DATA 0,0,0,0,0,0,4,4,4,4,0,0,0,0,0,0
118 kR DATA 0,0,0,4,4,4,3,4,4,4,2,4,4,0,0,0
119 qS DATA 0,3,4,3,4,4,4,4,4,4,4,4,2,4,0,0
120 TL DATA 3,4,3,4,3,4,4,4,4,2,4,2,4,2,4,2
121 u8 DATA 3,3,4,3,4,4,4,4,4,4,4,4,2,4,2,2
122 kU DATA 0,3,4,4,4,3,4,4,4,4,2,4,2,4,0,0
123 kR DATA 0,0,0,4,3,4,4,4,4,4,4,2,4,0,0,0
124 YE DATA 0,0,0,0,0,0,4,4,4,4,0,0,0,0,0,0
125 9n DATA 0,0,0,0,0,0,6,6,6,6,0,0,0,0,0,0
126 mL DATA 0,0,0,6,6,6,1,6,6,6,0,6,6,0,0,0
127 Ca DATA 0,1,6,1,6,6,6,6,6,6,6,6,0,6,0,0
128 jM DATA 1,6,1,6,1,6,1,6,6,0,6,0,6,0,6,0
129 Sx DATA 1,1,6,1,6,6,6,6,6,6,6,6,0,6,6,6
130 q0 DATA 0,1,6,6,6,1,6,6,6,6,0,6,0,6,0,0
131 yd DATA 0,0,0,6,1,6,6,6,6,6,6,6,6,0,0,0
132 Gu DATA 0,0,0,0,0,0,6,6,6,6,0,0,0,0,0,0
133 Z0 DATA 0,0,0,0,0,0,7,7,7,0,0,0,0,0,0,0
134 zt DATA 0,0,0,7,7,7,0,0,0,0,7,7,0,0,0,0
135 cl DATA 0,7,7,7,0,0,0,0,0,0,0,0,7,7,0,0
136 jr DATA 7,7,0,0,0,0,0,0,0,0,0,0,0,7,7,0
137 a0 DATA 7,7,0,0,0,0,0,0,0,0,0,0,0,7,7,0
138 fx DATA 0,7,7,7,0,0,0,0,0,0,0,0,7,7,0,0
139 7y DATA 0,0,0,7,7,7,0,0,0,0,7,7,0,0,0,0
140 gJ DATA 0,0,0,0,0,0,7,7,7,0,0,0,0,0,0,0
141 JF FOR s = 0 TO 3
142 LJ2 FOR y = 1 TO 8
143 OH4 FOR x = 1 TO 16
144 LG6 READ a:PSET(16*s+x,y),a
145 aYO NEXT x,y,s
146 xz GET (1,1)-(16,8),a1:GET (17,1)-(32,8),a2
147 i0 GET (33,1)-(48,8),a3:GET (49,1)-(64,8),a4
148 KM RETURN
149 Fo grafik:
150 os COLOR 1,0:CLS:z=1
151 dF FOR x = 30 TO 450 STEP 70
152 k22 FOR y = 20 TO 220 STEP 50
153 AK4 kx/y = x+RND(1)*18-GX:z = y+RND(1)*18-9
154 WM st(1,z)=0:z=z+1
155 iQ0 NEXT y,x
156 5a FOR x = 65 TO 415 STEP 70
157 q62 FOR y = 45 TO 195 STEP 50
158 My4 kx(z)=x+(RND(1)*26)-13:ky(z)=y+(RND(1)*26)-13
159 iR st(1,z)=0:z=z+1
160 nv0 NEXT y,x
161 oQ hh=1:1=1:sp2=0:sp1=26
162 ts FOR en = 1 TO 3:sp1=sp1+(6-en)
163 gD2 FOR x = 1 TO 59:FOR y = x TO 59
164 AN4 PSET(x,y),en
165 mS ss=SQD kx*x+ky*y : z = A: x = kx/5 : y = ky/5
166 KI IF (ss<sp1) AND (ss>sp2) THEN ve(hh)=x:ve(hh+1)=y:hh=hh+2:
167 J22 NEXT y,x
168 nb ha(en)=hh-2
169 oG0 NEXT en
170 yY GOSUB save:RETURN
171 ZL puty:
172 Qq FOR h = 1 TO 59
173 Jd2 IF POINT(kx(h),ky(h))=0 THEN put1
174 FL IF st(1,h)<2 THEN PUT (kx(h)-8,ky(h)-4),a4,PSET:GOTO put1
175 iD IF ro(h)>15 THEN PUT (kx(h)-8,ky(h)-4),a3,PSET:GOTO put1
176 H8 IF mo(h)>8 THEN PUT (kx(h)-8,ky(h)-4),a2,PSET:GOTO put1
177 UJ PUT (kx(h)-8,ky(h)-4),a1,PSET
178 bU0 put1:
179 iX NEXT h
180 Gs RETURN
181 7i manual:
182 v mm$(1)="Spiel laden ":mm$(2)="Neues starten "
183 sa mm$(3)="Welt kreieren ":mm$(4)="PRG - beenden "
184 94 ze = 4 GOSUB menu:RETURN
185 D5 menu2:
186 g8 mm$(1)="Bauen ":mm$(2)="Angriff "

```

```

187 VN z$ = "Transport":z$ = "Staats Info "
188 T4 mm$(5)="Planeten Info ":mm$(6)="Trafik "
189 FL mm$(7)="Revolution ":mm$(8)="Beenden "
190 NM me = 8:GOSUB menu:RETURN
191 Lb menu3:
192 fr mm$(1)="Spiel laden ":mm$(2)="Spiel sichern "
193 E7 me = 2:GOSUB menu:RETURN
194 Qh menu4:
195 WU mm$(1)="Neuanfang ":mm$(2)="In Ordnung "
196 sl mm$(3)="Beenden "
197 pG me=3:GOSUB menu:RETURN
198 Wo menu5:
199 of mm$(1)="Kriegsschiffe ":mm$(2)="Frachtschiffe "
200 Z2 mm$(3)="Geverbesentrum":mm$(4)="Ind.Zentrum "
201 vL mm$(5)="Weiter "
202 4X me=5:GOSUB menu:RETURN
203 Fe menu7:
204 67 mm$(1)="Save Game ":mm$(2)="Load Game "
205 9I mm$(3)="Ende ":mm$(4)="Weiter "
206 2j me=4:GOSUB menu:RETURN
207 i6 menu8:
208 rA mm$(1)="Entfernungen ":mm$(2)="Lager "
209 Jj mm$(3)="Transportech. ":mm$(4)="Kriegsschiffe "
210 qI mm$(5)="Planet ":mm$(6)="Weiter "
211 H0 me=6:GOSUB menu:RETURN
212 eb menu:
213 Ws COLOR 1,4
214 no LINE (506,2)-(630,240),7,bf:LINE (509,4)-(627,20),3,bf
215 F4 LINE (512,6)-(624,18),4,bf:LINE (510,20)-(626,20),2
216 vJ LINE (511,19)-(626,19),2:LINE (625,20)-(625,5),2
217 a5 LINE (626,20)-(626,4),2:LINE (627,20)-(627,4),2
218 Uk FOR t = 1 TO me
219 TB2 LOCATE 2,65:PRINT mm$(t)
220 aQ GET (510,4)-(626,20),a5
221 Jn LINE (510,4+t*18)-(626,20+t*18),0,bf
222 U6 PUT (510,4+t*18),a5
223 Op0 NEXT t
224 uB LOCATE 2,65:PRINT "Bitte wählen "
225 Xv wa=0:m=0:COLOR 1,0
226 gy LINE (512,16+wa*18)-(622,17+wa*18),1,bf
227 iA RETURN
228 WU zufall:
229 os FOR z = 1 TO 59:zu=RND(1)
230 bu2 IF zu > .75 THEN ro(z)=INT(RND(1)*10)+15 ELSE ro(z)=1+INT(RND(1)*10)
231 67 IF zu < .15 THEN mo(z)=INT(RND(1)*8)+8 ELSE mo(z)=1+INT(RND(1)*6)+2
232 gY st(z)=1:ro(z)=ro(z)*2*PND(1)*5+.9
233 zw tr(z)=0:ns(z)=0:re(pl)=0:la(pl)=0
234 y0 st(1,z)=0:st(2,z)=0:st(3,z)=0:st(4,z)=0
235 mJO NEXT z
236 Am RETURN
237 ms eingabe:
238 hi n=MOUSE(0)
239 zV IF n<=0 THEN eingabe
240 XD y=MOUSE(2):x=MOUSE(1)
241 2f IF x>507 THEN ein2
242 FQ z=0
243 Zs FOR h = 1 TO 59
244 i22 IF x<=kx(h)+10 AND x>=kx(h)-10 AND y<=ky(h)+5 AND y>=ky(h)-5 THEN m=0:s=hh+86
245 Mb0 NEXT h
246 E5 IF s = 0 THEN eingabe ELSE GOTO ein3:
247 ku ein2:
248 JW s=0:m=INT((y-4)/18):IF (m>me) OR (m<0) THEN GOTO eingabe
249 rY LINE (512,16+wa*18)-(622,17+wa*18),4,bf:LINE (512,16+m*18)-(622,17+m*18),1,bf:wa=m
250 e3 ein3:
251 P1 RETURN
252 JY viewiele:
253 oG COLOR 1,0:WINDOW 2,"",(200,100)-(440,150),2,1
254 JW PRINT:PRINT " - Wieviele Mitspieler - "
255 yx PRINT:PRINT " (1-4) ";:COLOR 7,0:INPUT spieler
256 vI IF spieler < 1 THEN spieler = 1
257 WF IF spieler > 4 THEN spieler = 4
258 uI spieler = INT(spieler)
259 8W FOR h = 1 TO spieler
260 ze2 COLOR 1,0:CLS:PRINT
261 7v PRINT " Name des ";h;". Spielers"
262 aD PRINT

```

Listing. »Fantasy« versetzt Sie in das Jahr 6533. Erobern Sie neue Welten.

```

263 OW PRINT " ";COLOR 7,0 INPUT na$(*)
264 w=0
265 eq=1
266 vy IF na$(u)<>0 THEN w=1
267 wk za=0
268 mp FOR r=1 TO ha(1):IF ve(r)=su THEN za=za+1
269 wT NEXT r
270 VT IF za<3 THEN w=1
271 dy na$(u)=st(h,su)+4:heim(h)=su:sh(su)=40
272 tv IF ro(su)<15 THEN la(su)=60 ELSE la(su)=ro(su)*2:sh(su)=10
273 FO IF mo(su)<10 THEN v(h)=80 ELSE v(h)=30:sh(su)=30
274 Ag f1(h)+1 f2(h)+2 f3(h)+1 f4(h)=10+NT RND(1)*5: f5(h)+1
275 VU IF ro(su)<8 THEN ro(su)=8
276 XS IF mo(su)<6 THEN mo(su)=6
277 s70 NEXT h
278 AK WINDOW CLOSE 2: RETURN
279 w werden
280 vO IF st(1,2)<2 OR st(1)>50 THEN BEEP:RETURN
281 al COLOR 1,7:LINE (64*8,24*8)-(630,28*8),7,bf
282 d1 LOCATE 25,65:PRINT "Schiffe senden",
283 6C LOCATE 26,65:PRINT "von NR.",z
284 fA LOCATE 27,65:PRINT "zu NR.",
285 36 z1=z
286 x1 GOSUB eingabe
287 xa IF (z=0) OR (z=s1) THEN LINE (64*8,24*8)-(630,28*8),7,bf:RETURN
288 8n PRINT z:COLOR 1,0
289 re er=SQR( (xx(2)-xx(z1))^2+(yy(2)-yy(z1))^2)
290 gZ IF er > er(f1(1)) THEN BEEP:RETURN
291 M2 GET (146,89)-(454,182),gfk
292 rY WINDOW 2,"Schiffe senden :", (150,100)-(450,180),0,1
293 NZ n$=" Von Planet NR.":GOSUB text:PRINT a1
294 12 PRINT " zu Planet NR.":z:PRINT
295 wh PRINT " Stationierte Kriegsschiffe ":sh(z):PRINT
296 10 PRINT " Wieviele ":PRINT " Kriegsschiffe ",
297 17 GOSUB of:INPUT a1:IF a1<0 THEN a1=0
298 SL IF a1>sh(z1) THEN a1=sh(z1)
299 dn del(1,1) z=heim(1,1):z1=1:del(1,1)=a1
300 94 sh z1=sr(21-a1,1,1)=sr(1,1)-1
301 k1 WINDOW CLOSE 2:PUT (146,89),gfk
302 Eq RETURN
303 ba bauen
304 EY p=heim(1):IF na(p)<>1 THEN RETURN
305 Ph GOSUB menu5
306 af bau1
307 IM GOSUB eingabe
308 nn bau2
309 2G IF wa=5 THEN GOSUB menu2:RETURN
310 71 ON wa GOTO krieg,trans,hand,indu
311 pZ GOTO bau1
312 Xa krieg:GET (146,89)-(404,192),gfk
313 F3 WINDOW 2,"Kriegsschiffe bauen :", (150,100)-(400,190),0,1
314 Ph f1=INT(la(p)/3) f2=INT(v(1)/3)
315 Yg GOSUB bau3
316 uU la p=la(p)-za*3 v(1)=v(1)-za*3:gl(1)=gl(1)+za
317 w WINDOW CLOSE 2:PUT (146,89),gfk:GOTO bau1
318 AD trans.
319 xH GET (146,89)-(404,192),gfk
320 v2 WINDOW 2,"Transportschiffe bauen :", (150,100)-(400,190),0,1
321 e6 f1=INT(la(p)/3) f2=INT(v(1)/3)
322 n GOSUB bau3
323 OW la p=la(p)-za*3 v(1)=v(1)-za*3:gl(1)=gl(1)+za
324 v WINDOW CLOSE 2:PUT (146,89),gfk:GOTO bau1
325 Hs bau3:n$=" Sie haben":GOSUB text:PRINT
326 rR PRINT " Geld für ";f2;" Schiffe und "
327 w PRINT " Rohstoffe für ";f1;" Schiffe."
328 ZK PRINT:PRINT " Wieviele Schiffe wollen Sie "
329 TE PRINT " bauen ";
330 Zv GOSUB of:INPUT sa:IF sa<0 THEN sa=0
331 YS IF (sa<f1) AND (sa<f2) THEN RETURN
332 n GOSUB sf:PRINT " Nicht genug Mittel !":PRINT:BEEP
333 r1 x=70:y=70:GOSUB ok za=C
334 KM RETURN
335 MC einkommen:
336 h6 GET (196,69)-(408,216),gfk
337 XO WINDOW 2,"Staats Info", (200,80)-(404,214),0,1
338 w C=0:t=0 g=0:a=0:q=0
339 17 FOR h=1 TO 59
340 BT2 IF na(h)=1 THEN g=g+mo(h):q=q+ro(h):a=a+la(h):C=C+sh(h):t=t+tr(h):
341 v90 NEXT h
342 Sb n$=" Besits ":GOSUB text

```

```

343 XT PRINT:PRINT:PRINT " Steuereinkommen ":g
344 KI PRINT " Vermoegen ":v(1):PRINT " Rohstoffabbau ",
345 RB PRINT " Rohstoffe ",s PRINT " Kriegsschiffe ",
346 4J PRINT " Transportschiffe ",t:PRINT
347 nd PRINT " Reichweite ",sr f1(1):PRINT " Bewaffnung ",
348 Jn PRINT " Panzerung ",f3(1)
349 EP x=43:y=115:GOSUB ok
350 ao WINDOW CLOSE 2:PUT (196,69),gfk
351 1d RETURN
352 DN Info
353 ed GET (176,89)-(420,220),gfk
354 S7 WINDOW 2,"Planeten Info", (180,100)-(416,216),0,1
355 In PRINT:n$=" Planet NR.":GOSUB text:PRINT z
356 79 PRINT:PRINT " Besitzer ":na$(na(s)):PRINT
357 Xq s=st(1,z):PRINT " Status ":st(s)
358 nH IF s=0 THEN infoend
359 's PRINT " Kriegsschiffe ":sh(s):PRINT " Einkommen ":mo(z)
360 y PRINT " Rohstoffe ",ro(z)
361 6t IF s=1 THEN infoend
362 5k PRINT " Lager ":la(z):PRINT " Transportschiffe ":tr(z)
363 2C infoend:
364 As x=01 y=99
365 w1 GOSUB ok
366 wO WINDOW CLOSE 2:PUT (176,89),gfk
367 Ht RETURN
368 Po w sav:
369 qH WINDOW 2,"Welt abspeichern", (200,100)-(440,150),2,1
370 A2 PRINT:PRINT " Bitte geben sie "
371 it PRINT " einen Namen ein":INPUT " ",a$
372 nH OPEN a$ FOR OUTPUT AS #1
373 gn PRINT#1,ha(1),ha(2),ha(3)
374 Cn FOR x=1 TO ha(3)+1:PRINT#1,ve(x):NEXT x
375 rB FOR x=1 TO 59:PRINT#1,kx(x),ky(x):NEXT x
376 vO CLOSE 1:WINDOW CLOSE 2
377 R3 RETURN
378 gO w lod
379 uC WINDOW 2,"Welt aussuchen", (200,100)-(440,150),2,1
380 re CLS:PRINT:PRINT " Bitte geben sie den "
381 k PRINT " Namen der Welt ein":PRINT " die sie benutzen wollen "
382 v8 INPUT " ",f$
383 u IF f$<>" " THEN GOSUB w lod
384 Gs WINDOW CLOSE 2:RETURN
385 vO w lod
386 uv OPEN f$ FOR INPUT AS #1
387 vI INPUT#1,ha(1),ha(2),ha(3)
388 49 FOR x=1 TO ha(3)+1:INPUT#1,ve(x):NEXT x
389 vU FOR x=1 TO 59:INPUT#1,kx(x),ky(x):NEXT x
390 6q CLOSE 1
391 nH RETURN
392 fr war:
393 w1 p=del(1,*)
394 X4 IF na(p)=1 THEN sh(p)=sh(p)+k1(1,v):RETURN
395 xB kr1
396 00 t1=k1(1,v):IF t1=0 THEN RETURN
397 Eb t2=sh(p)
398 4J kr2
399 Le f1=t1*((15-RND(1)*10)=(f2(1)-f3(1)))/100:IF f1<0 THEN f1=0
400 BD f2=t2*((15-RND(1)*10)=(f2(na p)-f3(na p)))/100:IF f2<0 THEN f2=0
401 f3 t1=INT(t1-f1):IF t1<0 THEN t1=0
402 wD t2=INT(t2-f2):IF t2<0 THEN t2=0
403 xK IF t1<t2 THEN ver1
404 RV IF t2<>0 THEN kr2
405 Ro sh(p)=t1:st(na(p),p)=1:na(p)=1
406 rS st(1,p)=2:IF p=heim(1) THEN st(1,p)=4
407 vE RETURN
408 zy ver1:IF na(heim(1,w))=1 THEN sh(heim(1,w))=sh(heim(1,w))+t1
409 fI sh(p)=t2:st(1,p)=1
410 ya RETURN
411 Rf trans1:
412 ss COLOR 1,7:LINE (64*8,24*8)-(630,28*8),7,bf
413 J8 LOCATE 25,65:PRINT "Transport "
414 DJ LOCATE 26,65:PRINT "von NR.",z
415 nH LOCATE 27,65:PRINT "zu NR.",
416 2D z1=z
417 48 GOSUB eingabe
418 4h IF (s=0) OR (z=z1) THEN LINE (64*8,24*8)-(630,28*8),7,bf:RETURN

```



```

URN
419 EN PRINT z:COLOR 1,0
420 L2 IF (st(1,s1)<2) OR (na(z)<>1) OR (na(s1)<>1) OR a2(1)
    >50 THEN BEEP:RETURN
421 zm er=SQ kx z kx z' 2 2-x'y'-ky(z1)'2
422 oh IF er > sr(f1(1)) THEN BEEP:RETURN
423 It GET (126,89)-(444,180),gf&
424 nt WINDOW 2,"Transportschiffe senden :", (130,100)-(440,178),0,
    1 GOSUB sf
425 Vh n$=" Von Planet NR.":GOSUB text:PRINT s1
426 9A PRINT " su Planet NR.":z PRINT
427 bs PRINT " Stationierte Transportschiffe ":tr(s1)
428 4c PRINT " Transportable Rohstoffe "
429 6M IF st(1,s1)<3 THEN l1=0:ELSE l1=1a(s1)
430 eM PRINT l1:IF st(1,s1)>2 THEN PRINT ""
431 ti PRINT " Wieviele Transportschiffe ":GOSUB of:INPUT a1:GO
    SUB sf
432 FO PRINT " Wieviele Rohstoffe ":GOSUB of:INPUT a2:GO
    SUB sf
433 hG IF a1>tr(s1) THEN a1=tr(s1)
434 5p IF a2>l1 THEN a2=l1
435 Im IF a2>a1 THEN a2=a1
436 q3 IF a1<0 THEN a1=0
437 38 IF a2<0 THEN a2=0
438 QZ de2(1,s2(1))=s:k2(1,s2(1))=a1:tr(s1)=tr(s1)-a1
439 5J x3(1,s2(1))=a2:a2(1,s2(1))=a2:s2(1)=s2(1)-1
440 Yg ens:
441 3t WINDOW CLOSE 2:PUT (126,89),gf&
442 36 RETURN
443 tm sf
444 36 CALL setfont (WINDOW(8),font2&):RETURN
445 1Y kh
446 nr LINE (64*8,24*8)-(630,28*8),7,bf
447 1V COLOR 1,7,LOCATE 25,65:BEEP:CALL setfont (WINDOW(8),font2&)
448 K3 PRINT "Kein Handel":LOCATE 26,65:PRINT "möglich":COLOR 1,0:
    RETURN
449 kd hand:
450 bf GOSUB eingabe
451 5A IF z=0 THEN GOTO bau2.
452 72 IF na(z)<>1 THEN hand
453 zr GET (196,69)-(444,162),gf&
454 wW WINDOW 2,"Handelszentrum bauen", (200,100)-(440,160),0,1
455 qk PRINT PRINT:GOSUB sf
456 FK IF st(1,z)>2 THEN n$=" Handelszentrum vorhanden":GOSUB tex
    t:GOTO han3
457 4X IF v(1)<50 THEN n$=" Zu wenig geld":GOSUB text:GOTO han2
458 gs v(1)=v(1)-50:st(1,z)=3
459 nr han1.
460 u7 n$=" Handelszentrum errichtet":GOSUB text
461 3a han2:
462 kB PRINT PRINT " Kosten : 50 "
463 17 han3.
464 yt PRINT " Vermögen ",v 1
465 oG x=60:y=36:GOSUB ok:WINDOW CLOSE 2:PUT (196,89),gf&:GOTO han
    d
466 Pk indu:
467 ew GOSUB eingabe
468 MR IF z=0 THEN GOTO bau2.
469 99 IF na(z)<>1 THEN indu
470 wW WINDOW 2,"Industriezentrum bauen", (180,100)-(440,160),2,1
471 60 PRINT:PRINT:GOSUB sf
472 4C IF st(1,z)>3 THEN n$=" Industriezentrum vorhanden":GOSUB t
    ext:GOTO ind3
473 3S IF v(1)<500 THEN n$=" Zu wenig geld":GOSUB text:GOTO ind2
474 1J v 1)=v 1)-50:st(1,heim(1))=3:st(1,z)=4:heim 1)=z
475 4C ind1:
476 Vt n$=" Industriezentrum errichte":GOSUB text
477 BK ind2
478 XQ PRINT:PRINT " Kosten : 500":PRINT " Vermögen : ",v(1)
479 IS ind3
480 w4 x=70:y=36:GOSUB ok:WINDOW CLOSE 2:GOSUB dis:GOTO indu
481 Ob nex
482 TQ WINDOW 2,"", (200,100)-(440,170),2,1
483 dK PRINT GOSUB sf:PRINT " Nächster Herrscher : "
484 HI PRINT:PRINT " ";na$(1):x=60:y=50:GOSUB ok:WINDOW CLOSE 2
485 Bn RETURN
486 ed erf.
487 IJ IF jahr<>f4(1) THEN RETURN
488 9p f4(1)=jahr+INT(RND(1)*20)-f5(1)*3:IF f4(1)<5+jahr THEN f4(
    1)=5+jahr
489 MD zu=RND(1):IF f1(1)=1 THEN zu=0
490 e4 IF (zu<.5) AND (f1(1)<3) THEN f1(1)=f1(1)+1:RETURN
491 Fv zu=RND(1)

```

```

492 FY IF zu<.4 THEN f3(1)=f3(1)+f5(1)-.9:RETURN
493 yX f2(1)=f2(1)+f5(1)-.9:RETURN
494 3J ok:
495 6C w=MOUSE(0):WINDOW OUTPUT 1:COLOR 1,4: LOCATE 2,65
496 V2 PRINT " weiter ":GET (510,4)-(626,20),a%
497 88 LOCATE 2,65: PRINT "Bitte wählen ":COLOR 1,0
498 rd WINDOW OUTPUT 2 PUT x,y,a% LINE (x+2,y-12)-(x+112,y+13),4
    ,bf
499 Q6 ok1:
500 1c w=MOUSE (0)
501 xK IF v<1 THEN ok1
502 34 RETURN
503 IX dis
504 7U LINE (0,0)-(505,240),0,bf
505 27 IF d1=0 THEN GOSUB puty:RETURN
506 B8 FOR h= 1 TO ha(d1) STEP 2:x=va(h):y=va(h+1)
507 Iq2 IF (st(1,x)>0) OR (st(1,y)>0) THEN LINE (kx(x),ky(x))-(
    x+1,y,ky(y)),1
508 hQ NEXT h
509 V2 GOSUB puty:CIRCLE (kx(heim(1))-1,ky(heim(1))-1),10,1,3,.05,
    .3
510 aC
511 47 rev:
512 1s IF h=1 THEN re(pl)=0:RETURN
513 0u WINDOW 2,"Revolution", (200,80)-(440,160),2,1:
514 4Q n$=" Auf Planet " GOSUB text:PRINT
515 67 PRINT " NR.",pl, " (",INT(kx pl /2),"/",INT(ky(pl)/2),")"
516 3C PRINT " fand eine Revol. oder statt.".PRINT
517 Kn re = INT((ro(pl)+mo(pl))/2+re(pl)/3)
518 Vy IF re<sh(pl) THEN PRINT " Sie wurde niedergemetzelt.":sh(
    pl)=INT(sh(pl)-re/3*2):GOTO reve
519 oz sh(pl)=INT(re-sh(pl)/3*2) tr(p)=0 st(na(pl),p)=1 na pl)=0
520 44 PRINT " Sie war erfolgreich."
521 1H reve
522 VO re pl,=0 x=60 y=58:GOSUB ox
523 V7 WINDOW CLOSE 2:RETURN
524 6I revo.
525 VJ IF st(1,z)<1 THEN RETURN:ELSE:GET (196,69)-(442,182),gf&
526 va WINDOW 2,"Revolution unterstützen", (200,80)-(440,180),0,1
527 6W n$=" Auf Planet NR.":GOSUB text:PRINT z
528 7p PRINT " Vermögen : ",v(1):PRINT PRINT " Wieviel ihres Ge
    ldes "
529 34 PRINT " wollen Sie investieren ":GOSUB of:INPUT g:GOSUB s
    f
530 Vw IF g>v(1) THEN gav(1)
531 55 IF g<0 THEN g=0
532 40 PRINT:PRINT " O.K. Vermögen : ",v(1)-g
533 6E v 1)=v 1)-g re z)=re z)-g
534 4G x=60 y=75:GOSUB ok:WINDOW CLOSE 2:PUT (196,69),gf&:RETURN
535 63 beenden
536 xG WINDOW 2,"Spiel Beenden !!!!! ", (200,80)-(440,140),2,1
537 4a n$=" Sind sie sicher ?":GOSUB text:PRINT
538 4a PRINT " (j/n) ":GOSUB of:INPUT s$
539 4V IF s$="j" OR s$="J" THEN WINDOW CLOSE 2:GOTO start
540 4G WINDOW CLOSE 2:GOTO wv6
541 20 sav
542 4n WINDOW 2,"Spielstand speichern", (200,80)-(440,140),2,1
543 3J n$=" Bitte geben sie einen":GOSUB text:PRINT
544 2J PRINT " Namen ein ", GOSUB of:INPUT s$:IF s$="" THEN savn
545 3d OPEN s$ FOR OUTPUT AS #3
546 4b PRINT#3,jahr,spieler,#3
547 1a FOR p= 1 TO 59
548 47 PRINT#3,mo(p),sh(p),tr(p),ro(p),na(p),re(p),la(p)
549 41 NEXT
550 72 FOR y = 1 TO spieler
551 4H2 FOR x= 1 TO 59 PRINT#3,st(y,x)
552 4HO NEXT x,y
553 4V FOR x= 1 TO spieler
554 4a2 PRINT#3,na$(x)
555 4N PRINT#3,heim(x),f1(x),f2(x),f3(x),f4(x),f5(x),v(x)
556 4OO NEXT x
557 4f CLOSE 3
558 BK savn.
559 4t WINDOW CLOSE 2
560 30 RETURN
561 6p lod.
562 4W WINDOW 2,"Spielstand laden", (200,80)-(440,140),2,1
563 8Q n$=" Bitte geben sie der " GOSUB text
564 4E PRINT:PRINT " Namen des Spiels ein":GOSUB of
565 4T INPUT " s$:s$="" THEN lode

```

**Listing. »Fantasy« versetzt Sie in das Jahr 6533.
Erobern Sie neue Welten (Fortsetzung).**

```

566 Me OPEN s$ FOR INPUT AS #3
567 me INPUT #3, jahr, spieler, f$
568 au FOR p= 1 TO 59
569 o62 INPUT #3, mo(p), sh(p), tr(p), ro(p), na(p), re(p), la(p)
570 nro NEXT p
571 o6 FOR y = 1 TO spieler
572 o62 FOR x= 1 TO 59: INPUT #3, st(y,x)
573 u60 NEXT y
574 o6 FOR x= 1 TO spieler
575 wk2 INPUT #3, na$(x), heim(x), f1(x), f2(x), f3(x), f4(x), f5(x), v(x)
576 .10 NEXT x
577 De CLOSE 3
578 VP GOSUB wmlod
579 57 lode:
580 YE WINDOW CLOSE 2
581 jL RETURN
582 Oy graf:
583 aN GOSUB menu8
584 Oy graf1:
585 mq GOSUB eingabe
586 JP IF w<1 OF a> THEN GOSUB menu2 RETURN
587 Mk ON = GOSUB dars, p1, p2, p3, puty: CIRCLE (kx(heim(1))-1, ky(heim(1))-1), 10, 1, 3, .05, .3
588 .R GOTO graf1
589 FY pro:
590 4I LINE (0,0)-(505,240), 0, bf
591 NV IF d1=0 THEN GOSUB puty: RETURN
592 2w FOR h= 1 TO ha(d1) STEP 2: x=ve(h): y=ve(h+1)
593 r22 IF (st(1,x)>0) OR (st(1,y)>0) THEN LINE (kx(x), ky(x))-(kx(y), ky(y)), 1:
594 .50 NEXT h
595 en ON = GOSUB p1, p1, p2, p3, puty: CIRCLE (kx(heim(1))-1, ky(heim(1))-1), 10, 1, 3, .05, .3
596 .a RETURN
597 ih p1:
598 i1 FOR h= 1 TO 59
599 o.2 IF POINT kx h, ky = 0 THEN pt1
600 28 IF st(1,h)<2 THEN PUT (kx(h)-8, ky(h)-4), a4, PSET: GOTO pt1
601 49 IF la(h)>150 THEN PUT (kx(h)-8, ky(h)-4), a3, PSET: GOTO pt1
602 h1 IF la(h)>30 THEN PUT (kx(h)-8, ky(h)-4), a2, PSET: GOTO pt1
603 MB PUT (kx(h)-8, ky(h)-4), a1, PSET
604 nvo pt1
605 AP NEXT
606 3k RETURN
607 v7 p2:
608 5s FOR h= 1 TO 59
609 la2 IF POINT(kx(h), ky(h))=0 THEN pt2
610 ET IF st(1,h)<2 THEN PUT (kx(h)-8, ky(h)-4), a4, PSET: GOTO pt2
611 3N IF tr(h)>50 THEN PUT (kx(h)-8, ky(h)-4), a3, PSET: GOTO pt2
612 3D IF tr(h)>10 THEN PUT (kx(h)-8, ky(h)-4), a2, PSET: GOTO pt2
613 eL PUT (kx(h)-8, ky(h)-4), a1, PSET
614 oAO pt2:
615 K2 NEXT
616 .a RETURN
617 89 p3
618 r2 ON = 1 TO 59
619 o62 IF POINT(kx(h), ky(h))=0 THEN pt3
620 .8 IF st(1,h)<2 THEN PUT (kx(h)-8, ky(h)-4), a4, PSET: GOTO pt3
621 39 IF sh(h)>50 THEN PUT (kx(h)-8, ky(h)-4), a3, PSET: GOTO pt3
622 f2 IF sh(h)>10 THEN PUT (kx(h)-8, ky(h)-4), a2, PSET: GOTO pt3
623 gV PUT (kx(h)-8, ky(h)-4), a1, PSET
624 2P0 pt3:
625 3J NEXT
626 34 RETURN
627 r1 font:
628 7k name0$="sapphire.font"+CHR$(0)
629 2k .5 = 1
630 11 textattr$(0) = SADD(name0$): textattr$(1) = h$*2^16+0*2^4+0
631 6e font1$ = openfont$(VARPTR(textattr$(0)))
632 0v IF font1$=0 THEN font1$ = opendiskfont$(VARPTR(textattr$(0)))
633 5g name0$="opal.font"+CHR$(0): h$ = 9
634 1m textattr$(0) = SADD(name0$): textattr$(1) = h$*2^16+0*2^4+0
635 6k font2$ = openfont$(VARPTR(textattr$(0)))
636 78 IF font2$=0 THEN font2$ = opendiskfont$(VARPTR(textattr$(0)))
637 90 name0$="topaz.font"+CHR$(0): h$ = 8
638 7q textattr$(0) = SADD(name0$): textattr$(1) = h$*2^16+0*2^4+0
639 8k font0$ = openfont$(VARPTR(textattr$(0)))
640 7f name0$="diamond.font"+CHR$(0): h$ = 20
641 w textattr$(0) = SADD(name0$): textattr$(1) = h$*2^16+0*2^4+0
642 1t font3$ = openfont$(VARPTR(textattr$(0)))
643 n0 IF font3$=0 THEN font3$ = opendiskfont$(VARPTR(textattr$(0)))

```

```

))
644 mN RETURN
645 1q of:
646 2t CALL setfont (WINDOW(8), font0$): RETURN
647 S1 text:
648 2i CALL setfont (WINDOW(8), font1$): CALL move(WINDOW(8), 1, 13)
649 X0 COLOR 7, 0: PRINT LEFT$(n$, 2): CALL setfont (WINDOW(8), font2$)
650 Kw COLOR 1, 0: PRINT RIGHT$(n$, LEN(n$)-2):
651 rT RETURN
652 QA cfont:
653 Ak CALL closefont (font1$): CALL closefont (font2$)
654 3n CALL closefont (font3$): CALL closefont (font0$)
655 vK RETURN
656 Ua sieg.
657 UI GOSUB titell
658 tI FOR x=1 TO spieler: IF na(heim(x))=x THEN wi=x
659 Y3 NEXT x
660 XD PALETTE 4, 0, 0, 1: PALETTE 5, 0, 0, .5: x%=90: y%=120: COLOR 6, 0
661 1K CALL setfont (WINDOW(8), font2$): POKER WINDOW(8)+64, 1
662 D3 na$=SetSoftStyle$(WINDOW(8), 2, en$)
663 SW "AI" move(WINDOW(8), x%, y%, 1) PRINT "Gewonnen hat " COLOR 7 0
664 Vg CALL move(WINDOW(8), x%+1, y%+1): PRINT "Gewonnen hat : "
665 y8 x%=70: y%=150: COLOR 4, 0: CALL setfont (WINDOW(8), font3$)
666 t8 POKER WINDOW(8)+64, 2: na$=SetSoftStyle$(WINDOW(8), 4, en$)
667 o7 CALL move(WINDOW(8), x%, y%): PRINT na$(vi): COLOR 5, 0
668 40 CALL move(WINDOW(8), x%+1, y%+1): PRINT na$(vi)
669 9a ttu:
670 1M w=MOUSE (0)
671 KX IF w<1 THEN ttu
672 Y1 CALL SetDrMd(WINDOW(8), 1)
673 AD WINDOW CLOSE 6 : SCREEN CLOSE 2
674 my GOSUB titel: GOSUB srr: GOTO neu8
675 Jp titel:
676 MP GOSUB titel1: GOTO titel2
677 L9 titel2:
678 wI x%=60: y%=120: COLOR 4, 0: CALL setfont (WINDOW(8), font1$)
679 Y9 "x% = 60, y% = 120, 1 CALL move(WINDOW(8), x%, y%)
680 3G PRINT "Written in 1988 by: " COLOR 5, 0
681 62 CALL move(WINDOW(8), x%+1, y%+1): PRINT "Written in 1988 by: "
682 Je x%=50: y%=160: COLOR 6, 0: CALL setfont (WINDOW(8), font3$)
683 K5 POKER WINDOW(8)+64, 2: na$=SetSoftStyle$(WINDOW(8), 4, en$)
684 Ut CALL move(WINDOW(8), x%, y%): PRINT "Holger Hoffmann": COLOR 7, 0
685 50 CALL move(WINDOW(8), x%+1, y%+1): PRINT "Holger Hoffmann"
686 ex tti:
687 2d w=MOUSE (0)
688 RQ IF w<1 THEN tti
689 pe CALL SetDrMd(WINDOW(8), 1)
690 R5 WINDOW CLOSE 6 : SCREEN CLOSE 2: RETURN
691 XE titell1:
692 47 SCREEN 2, 320, 256, 3, 1: WINDOW 6, "", (0,0)-(310,240), 0, 2
693 8D CALL setfont (WINDOW(8), font3$): en$=AskSoftStyle$(WINDOW(8))
694 Hf na$=SetSoftStyle$(WINDOW(8), 2, en$) CALL SetDrMd WINDOW 8 0
695 1h POKER WINDOW(8)+64, 4: PALETTE 0, 0, 0, 0: PALETTE 1, 0, 0, 0
696 6t PALETTE 2, 1, 0, 1: PALETTE 3, 0, 0, 0: PALETTE 4, 0, 1, 0
697 39 PALETTE 5, 0, 0, 0: PALETTE 6, 0, 0, 0: PALETTE 7, 0, 0, 0
698 8b x%=62: y%=65: COLOR 2, 0: CALL move(WINDOW(8), x%, y%)
699 0g PRINT " FANTASY " COLOR 3, 0: CALL move(WINDOW(8), x%+1, y%+1)
700 9H PRINT " FANTASY ": LINE (0,73)-(318,73), 2
701 8a LINE (1,74)-(319,74), 3
702 gI RETURN
703 41 srr:
704 12 SCREEN 1, 640, 256, 3, 2
705 gI WINDOW 1, "Fantasy Written in 1988 by Holger Hoffmann", 0, 1
706 32 MENU 1, 0, 1, "": MENU 2, 0, 1, "": MENU 3, 0, 1, "": MENU 4, 0, 1, ""
707 8 RETURN
708 1 Fehler:
709 qz WINDOW 2, "Fehlermeldung", (200,100)-(360,175), 2, 1
710 WB n$=" Error !!!!!-GOSUB text
711 NE PRINT: PRINT IF ERR<>53 THEN PRINT " Fehler Nr.": ERR: ELSE PRINT " File not found"
712 2i PRINT: PRINT " Mit Return": PRINT " bestätigen ": COLOR 0, 0: INPUT n$
713 re WINDOW CLOSE 2: RESUME start
714 K5 REM Ende von FANTASY
715 r7 REM Written in 1988 by Holger Hoffmann
(C) 1989 M&T

```

Listing. »Fantasy« versetzt Sie in das Jahr 6533.
Erobere Sie neue Welten (Schluß).

Trainingslager...

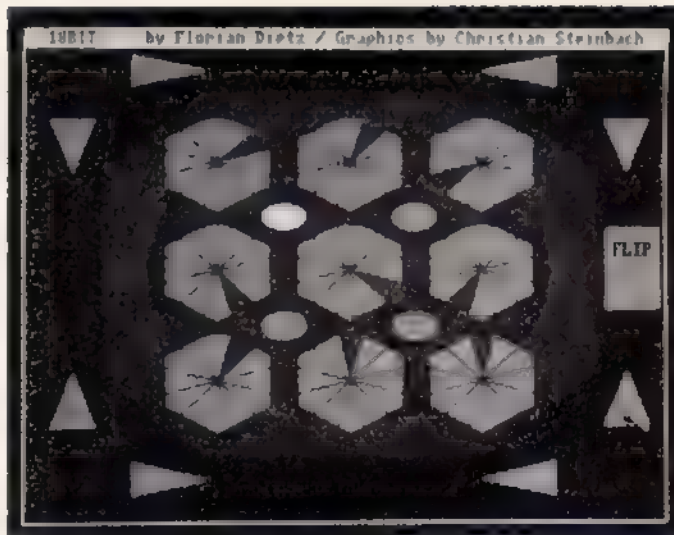
Muskeln stählt man im Bodybuilding-Center mit Hanteln. Die »Hanteln«, um das Gehirn zu trainieren, sind Denkspiele. Folgen Sie uns jetzt in den Übungsraum.

Unsere Übung heißt »18-BIT« und fördert das räumliche Vorstellungsvermögen. Im Bild sehen Sie das Trainingsgerät. Es handelt sich um neun Zeiger und vier Knöpfe. Die Aufgabe ist es, alle Zeiger so zu drehen, daß sie nach oben zeigen. Sie glauben, das sei einfach? Weit gefehlt, denn die Zeiger sind untereinander verbunden. Außerdem befinden sich auf der »Rückseite« des achteckigen Feldes noch einmal neun Zeiger. Natürlich müssen auch diese nach oben zeigen. Es sind also insgesamt 18 Zeiger, woher auch der Name des Spiels stammt. Die zwei Seiten sind farblich (blau und grün) unterschieden. Die Rückseite sehen Sie, wenn Sie den Schalter mit der Aufschrift »FLIP« anklicken.

Um den angestrebten Zustand zu erreichen, stehen Ihnen an den vier Eckpunkten acht Schalter zur Verfügung. Mit ihnen lassen sich die Anzeigen verstellen. Je nach Pfeilrichtung im oder gegen den Uhrzeigersinn. Jetzt kommt das räumliche Vorstellungsvermögen zum Zug. Die Anzeigen auf der Vorder- und Rückseite sind ebenfalls miteinander verbunden. Dreht man die rechte untere Anzeige nach links, bewegt sich der entsprechende Zeiger (Rückseite links unten) entgegengesetzt. Welche Anzeigen sich automatisch mitdrehen, läßt sich über die vier Knöpfe zwischen den Anzeigen beeinflussen. Gelbe Knöpfe sind aktiv, lila Knöpfe sind inaktiv. Auf der anderen Seite ist der Zustand der Knöpfe genau umgekehrt. Wieder sind die Positionen seitenverkehrt.

Die Einstellung der Knöpfe ist entscheidend für das Verhalten der Anzeigen. Sind alle Knöpfe aktiv, drehen sich alle Zeiger mit, egal an welcher Ecke gedreht wird. Stellen Sie alle Knöpfe ab, bewegen sich die vier Scheiben an den Ecken des Spielfeldes. Ein Knopf stellt also immer die Verbindung zwischen den vier ihn umgebenden Anzeigen her. Wenn Sie jetzt den Eindruck haben, daß 18BIT schwierig zu lösen ist, haben Sie recht. Möglich ist es aber auf jeden Fall. Nach längerem Spiel bekommt man langsam ein Gefühl für das Problem.

Außer der Bedienung mit den Schaltern besitzt 18BIT noch drei Pull-Down-Menüs für andere Funktionen, die wir Ihnen in den folgenden Punkten ausführlich vorstellen wollen:



Die eine Seite von »18BIT« mit neun Zeigern und vier Schaltern. Finden Sie eine Lösung?

alte mit »Quicksave« gespeicherte Spielstand wird überschrieben.

■ Project/Load

Gespeicherte Spielstände können geladen werden. Vor dem Laden fragt 18BIT nach dem Dateinamen. Wie bei allen anderen Dateifunktionen ist es möglich, einen ganzen Pfad anzugeben.

■ Project/Quickload

Lädt den Spielstand in der Datei »18BIT.quick«.

■ Project/Quit

Beendet das Spiel, verläßt aber nicht den Amiga-Basic-Interpreter.

18BIT bietet Ihnen eine Hilfe, die Sie sicher bald nicht mehr missen wollen: Sequenzen. Wenn Sie für eine Problemstellung einen Lösungsweg gefunden haben, können Sie diese Sequenz auf Diskette speichern und später wieder abrufen.

■ Edit/SeqStart

Hiermit schalten Sie 18BIT auf »Aufnahme«. Das Programm merkt sich von nun ab alle Drehungen, Seitenwechsel und Knopf-einstellungen. Vor der Eingabe der Spielzüge verlangt 18BIT noch einen Namen für die anzulegende Datei.

■ Edit/SeqStop

Beendet den Aufnahmemodus und speichert die Daten in der Datei mit dem angegebenen Namen.

■ Edit/SeqShow

Fragt nach dem Namen der Datei und lädt die Sequenz, die dann automatisch durchgeführt wird.

■ System/Sound On

Schaltet den Ton bei Drehung der Anzeigen an. Durch das akustische Signal ist es leichter, zu verfolgen, wieviele Anzeigen verstellt werden.

■ System/Sound Off

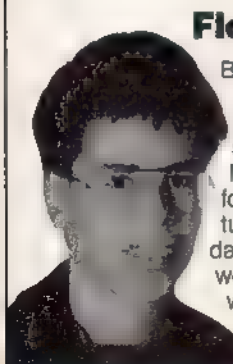
Der Sound wird ausgeschaltet, was bewirkt, daß das Spiel etwas schneller läuft.

■ System/Cycling On

18BIT verändert die Farben der Kreise, wodurch der Eindruck von Bewegung entsteht. Beim Verstellen von Zeigern

Florian Dietz/Christian Steinbach

Beide Autoren von 18BIT besuchen die 13. Klasse des Uhland-Gymnasiums in Tübingen. Die Freundschaft zwischen beiden begann im Jahr 1986, in dem sie auch je einen C64 bekamen. Nach Basic folgte die Programmierung in Comal. Vor ca. eineinhalb Jahren erfolgte dann der Aufstieg zum Amiga. Nach der Einarbeitung in Amiga-Basic entstand in den Weihnachtsferien 88/89 das Programm des Monats. Jetzt lernen die beiden C. Ein weiteres Projekt ist schon in Vorbereitung. Die Redaktion wünscht beiden, daß dies ein ebenso erfolgreiches Programm wird wie das letzte.



oder bei einem Seitenwechsel wird das Colorcycling (Farbrotation) kurz angehalten.

■ System/Cycling Off

Ohne sich verändernde Farben läuft 18BIT nicht schneller. Für die Augen ist diese Einstellung bei längerem Spiel angenehmer.

18BIT läßt sich mit dem AC-Basic-Compiler übersetzen. Die Schalter »N« und »R« müssen dabei eingeschaltet sein.

Jetzt macht »Gehirntraining« wirklich Spaß. Passen Sie auf, daß Sie keinen Muskelkater bekommen.

René Beaupol/kn

Programmname: 18BIT

Computer: A500, A1000, A2000 m Kickstart 1.2 u. 1.3

Sprache: Amiga-Basic 1.2

Bemerkung: siehe Text

Programmsautor: Florian Dietz Christian Steinbach

```

1 730 REM ** 18BIT Programmed by Florian Dietz
2 0 REM ** Graphics by Christian Steinbach
3 1 I EAP , 2000 , 5
4 1 I S I E I a p p r o g r a m m
5 1 k S M
6 1 E REM Hauptprogramm
7 1 t HauptProgramm:
8 1 m2 GOSUB VariableDefining
9 1 A GOSUB OutputDefinition
10 1 t GOSUB MenuDefining
11 1 f ON MENU GOSUB MenuTrapping
12 1 e ON MOUSE GOSUB MouseTrapping
13 1 m0 ON BREAK GOSUB BreakTrapping
14 1 n2 MOUSE STOP
15 1 n4 MENU STOP
16 1 F BREAK STOP
17 1 G GOSUB Graphic
18 1 G MOUSE STOP
19 1 y MENU STOP
20 1 x GOSUB PositionPaint
21 1 G Cycle=1
22 1 x GOSUB Blue
23 1 2 MOUSE ON
24 1 2 MENU ON
25 1 e BREAK ON
26 1 d WHILE =1
27 1 s TIME
28 1 o0 RETURN
29 1 x REM Belegung und Dimensionierung von Variablen und Feldern
30 1 W VariableDefining:
31 1 F TIMER ON
32 1 X OPTION BASE 1
33 1 u DIM Display(3,3,2), OldDisplay(3,3,2)
34 1 s DIM Button$(4), Button(4,2)
35 1 c DIM Flipped(3,3,2)
36 1 w DIM xpos(12), ypos(12)
37 1 m2 DIM xpos2(6), ypos2(6)
38 1 t RESTORE SectorData
39 1 y FOR x=1 TO 12
40 1 74 READ xpos(x), ypos(x)
41 1 52 NEXT x
42 1 x RESTORE SectorData2
43 1 1 FOR x=1 TO 6
44 1 04 READ xpos2(x), ypos2(x)
45 1 42 NEXT x
46 1 e Button$(1)="122111"
47 1 x Button$(2)="213231"
48 1 F Button$(3)="211111"
49 1 x Button$(4)="322333"
50 1 t FOR x=1 TO 4
51 1 04 FOR y=1 TO 2
52 1 86 Button(x,y)=y-1
53 1 k4 NEXT y
54 1 n2 NEXT x
55 1 d RANDOMIZE TIMEP
56 1 7 FOR x=1 TO 3
57 1 74 FOR y=1 TO 3
58 1 x6 FOR z=1 TO 2
59 1 43 Display(x,y,z)=INT(1+12*RND)
60 1 2 OldDisplay(x,y,z)=Display(x,y,z)
61 1 46 NEXT z
62 1 74 NEXT y
63 1 72 NEXT x

```



```

64 1 0 I FOR x=1 TO 3 STEP 2
65 1 14 FOR y=1 TO 3 STEP 2
66 1 x6 Display(x,y,z)=12-Disp(x,y,z)
67 1 4 IF Display(x,y,z)=0 THEN Display(x,y,z)=12
68 1 24 NEXT y
69 1 x2 NEXT x
70 1 7 RANDOMIZE TIMEP
71 1 0 DIM Welle$(256)
72 1 m FOR x=0 TO 255
73 1 84 Welle$(x)=1+RND*255-128
74 1 02 NEXT x
75 1 8 WAVE=0,Welle$
76 1 ERASE Welle$
7 1 5 DIM Fa(11)
78 1 1 Farbe=1
79 1 8 FOR x=1 TO 9
80 1 74 Fa(x)=Farbe-1/16
81 1 L Farbe=Farbe-1/16
82 1 2 NEXT x
83 1 k Filename$=""
84 1 8 Oldname$=""
85 1 1 J And1=1
86 1 m0 SETKEY
87 1 3 REM Öffnen eines neuen Screens und Neudefinition des Amiga
beiwindows:
88 1 q OutputDefinition:
89 1 2 SCREEN 1,520,240,4,2
90 1 w WINDOW 1," 18BIT . by Florian Dietz / Graphics by Chri
tian Steinbach",16,1
91 1 0 RETURN
92 1 7 REM Belegung der Menues und Start des Event Trappings fuer
die Menues
93 1 v1 MenuDefining:
94 1 02 MENU 1,0,1,"Project"
95 1 2 MENU 1,1,1,"New"
96 1 4 MENU 1,2,1,"Save"
97 1 7 MENU 1,3,1,"Save As"
98 1 5 MENU 1,4,1,"Quicksave"
99 1 3 MENU 1,5,1,"Load"
100 1 1 MENU 1,6,1,"Quickload"
101 1 d MENU 1,7,1,"Quit"
102 1 A MENU 2,0,1,"Edit"
103 1 m MENU 2,1,1,"SeqStart"
104 1 6 MENU 2,2,1,"SeqStop"
105 1 6 MENU 2,3,1,"SeqShow"
106 1 C MENU 3,0,1,"System"
107 1 s MENU 3,1,2," Sound On "
108 1 4 MENU 3,2,1," Sound Off "
109 1 B MENU 3,3,2," Cycling On "
110 1 J MENU 3,4,1," Cycling Off"
111 1 m MENU 4,0,0,""
112 1 m0 RETURN
113 1 7 REM Verzweigung zu den einzelnen Menuepunkten
114 1 e MenuTrapping
115 1 x32 MOUSE STOP
116 1 x MENU STOP
117 1 j Men=MENU(1)
118 1 m Punkt=MENU(1)
119 1 d IF Men=1 THEN ON Punkt GOSUB New1,Save1,SaveAs1,Quicksave
1,Load1,Quickload1,Quit1
120 1 w IF Men=2 THEN ON Punkt GOSUB SeqStart1,SeqStop1,SeqShow1
121 1 k IF Men=3 THEN ON Punkt GOSUB SoundOn1,SoundOff1,CyclingOn
1,CyclingOff1
122 1 x MOUSE ON
123 1 d MENU ON
124 1 y0 RETURN
125 1 2 REM Auswertung der Maussteuerung:
126 1 0 MouseTrapping
127 1 x2 LTaste=MOUSE(0)
128 1 w x=MOUSE(1)
129 1 8 y=MOUSE(2)
130 1 I MOUSE STOP
131 1 m MENU STOP
132 1 f IF x>466 AND x<511 AND y>88 AND y<129 THEN GOSUB Sere
enChange
133 1 0 IF LTaste<>0 AND WINDOW(0)=1 AND Seite=0 THEN
134 1 24 IF x>84 AND x<145 AND y>1 AND y<18 THEN Richtung=1
: Hebel=1 : GOSUB DisplayPosition
135 1 d IF x>18 AND x<57 AND y>34 AND y<61 THEN Richtung=-1
: Hebel=1 : GOSUB DisplayPosition
136 1 x IF x>366 AND x<427 AND y>1 AND y<18 THEN Richtung=-
1 : Hebel=2 : GOSUB DisplayPosition

```



```

137 xF IF x>464 AND x<503 AND y>34 AND y<61 THEN Richtung=
138 rY 1 : Hebel=2 : GOSUB DisplayPosition
139 Hq IF x>84 AND x<145 AND y>199 AND y<216 THEN Richtung
140 w= -1 : Hebel=3 : GOSUB DisplayPosition
141 Gu IF x>18 AND x<57 AND y>157 AND y<184 THEN Richtung=
142 Za 1 : Hebel=3 : GOSUB DisplayPosition
143 yu6 IF x>366 AND x<427 AND y>199 AND y<216 THEN Richtung
144 G94 g=1 : Hebel=4 : GOSUB DisplayPosition
145 b0 IF x>464 AND x<503 AND y>157 AND y<184 THEN Richtung
146 J76 g=-1 : Hebel=4 : GOSUB DisplayPosition
147 J24 IF x>190 AND x<227 AND y>76 AND y<90 THEN
148 Ad Button(1,1)=1-Button(1,1) : Button(1,2)=1-Button(1,1)
149 iJ6 : PAINT (207,83),2+Button(1,1),13
150 MP4 END IF
151 Cg IF x>294 AND x<331 AND y>76 AND y<90 THEN
152 Fx6 Button(2,1)=1-Button(2,1) : Button(2,2)=1-Button(2,1)
153 P14 : PAINT (312,83),2+Button(2,1),13
154 EP2 END IF
155 QQ4 IF x>190 AND x<227 AND y>129 AND y<143 THEN
156 Aa Button(3,1)=1-Button(3,1) : Button(3,2)=1-Button(3,1)
157 by : PAINT (207,136),2+Button(3,1),13
158 AS END IF
159 Ko IF x>294 AND x<331 AND y>129 AND y<143 THEN
160 Jn Button(4,1)=1-Button(4,1) : Button(4,2)=1-Button(4,1)
161 li : PAINT (312,136),2+Button(4,1),13
162 Fv END IF
163 uI ELSEIF LTaste<>0 AND WINDOW(0)=1 AND Seite=1 THEN
164 eb6 IF x>84 AND x<145 AND y>1 AND y<18 THEN Richtung=-1
165 bU4 : Hebel=2 : GOSUB DisplayPosition
166 wI IF x>18 AND x<57 AND y>34 AND y<61 THEN Richtung=1
167 RJ6 : Hebel=2 : GOSUB DisplayPosition
168 ex4 IF x>366 AND x<427 AND y>2 AND y<19 THEN Richtung=1
169 y IF x>464 AND x<503 AND y>34 AND y<61 THEN Richtung=
170 bS6 -1 : Hebel=1 : GOSUB DisplayPosition
171 ha4 IF x>84 AND x<145 AND y>199 AND y<216 THEN Richtung
172 X1 =1 : Hebel=4 : GOSUB DisplayPosition
173 H66 IF x>18 AND x<57 AND y>157 AND y<184 THEN Richtung=
174 kd4 -1 : Hebel=4 : GOSUB DisplayPosition
175 le2 IF x>366 AND x<427 AND y>200 AND y<217 THEN Richtung
176 vC g=-1 : Hebel=3 : GOSUB DisplayPosition
177 vU IF x>464 AND x<503 AND y>157 AND y<184 THEN Richtung
178 E30 g=1 : Hebel=3 : GOSUB DisplayPosition
179 hM IF x>190 AND x<227 AND y>76 AND y<90 THEN
180 E8 ColourDefining.
181 fP2 Button(2,1)=1-Button(2,1) : Button(2,2)=1-Button(2,1)
182 MU : PAINT (207,83),2+Button(2,2),13
183 Yr4 END IF
184 rn IF x>294 AND x<331 AND y>76 AND y<90 THEN
185 J2 Button(1,1)=1-Button(1,1) : Button(1,2)=1-Button(1,1)
186 ja : PAINT (312,83),2+Button(1,2),13
187 cJ4 END IF
188 vt IF x>190 AND x<227 AND y>129 AND y<143 THEN
189 yT2 Button(4,1)=1-Button(4,1) : Button(4,2)=1-Button(4,1)
190 Q20 : PAINT (207,136),2+Button(4,2),13
191 Ag END IF
192 2Q IF x>294 AND x<331 AND y>129 AND y<143 THEN
193 Oe2 Button(3,1)=1-Button(3,1) : Button(3,2)=1-Button(3,1)
194 Se : PAINT (312,136),2+Button(3,2),13
195 E80 END IF
196 v5 REM Farbfestlegung fuer die Grundfarben beider Seiten
197 WY2 REM ColourDefining.

```

```

198 W74 FOR x=12 TO 4 STEP -1
199 J6 PALETTE x,0,0,Pa(x-3),
200 e4 NEXT x
201 S-2 END IF
202 7 WHILE Cycle=1
203 C4 GOSUB Cycling
204 b2 FOR x=12 TO 4 STEP -1
205 -6 PALETTE x,0,0,Pa(x-3)
206 -34 NEXT x
207 -h MOUSE ON
208 e2 MENU ON
209 C2 MOUSE STOP
210 v3 MENU STOP
211 -e2 REM
212 -80 REM
213 7 REM Farbfestlegung fuer die grueene/zweite Seite
214 -e green
215 -e2 IF Cycle=0 THEN
216 -24 FOR x=12 TO 4 STEP -1
217 -36 PALETTE x,0,Pa(x-3)-1/8,0
218 -44 NEXT x
219 -44 END IF
220 -24 WHILE Cycle=1
221 -34 GOSUB Cycling
222 -7 FOR x=12 TO 4 STEP -1
223 -96 PALETTE x,0,Pa(x-3)-1/8,0
224 -124 NEXT x
225 -24 MOUSE ON
226 -24 MENU ON
227 -24 MOUSE STOP
228 -24 MENU STOP
229 -24 REM
230 -24 RETURN
231 -24 REM Colour Cycling-Routine fuer beide Seiten:
232 -24
233 -12 Pa(10)=Pa(1)
234 -12 FOR x=1 TO 9
235 -14 Pa(x)=Pa(x+1)
236 -12 NEXT x
237 -24 RETURN
238 -24 REM Redefinierung der Ausgaben:
239 -24 Quit1:
240 -24 TIME OFF
241 -24 MOUSE OFF
242 -24 MENU OFF
243 -24 MENU RESET
244 -24 END
245 -24 REM
246 -24 REM Variablen- und Screenwechsel auf die andere Seite.
247 -24 ScreenChange.
248 -24 Seite=1-Seite
249 -24 GOSUB PositionChange
250 -24 RETURN
251 -24 REM Berechnung der Displaystellungen:
252 -24 DisplayPosition:
253 -24 IF Seq=1 THEN GOSUB Sequence
254 -24 FOR x=1 TO 3
255 -24 FOR y=1 TO 3
256 -24 OldDisplay(x,y,Seite+1)=Display(x,y,Seite+1)
257 -24 NEXT y
258 -24 NEXT x
259 -24 AB=Button(Hebel,2)+1
260 -24 IF AB=2 THEN Richtung=-Richtung
261 -24 FOR x=1 TO 4
262 -24 IF Button(x,AB)=1 THEN
263 -24 FOR y=1 TO 3
264 -24 x$=MID$(Button$(x),2*y-1,1)
265 -24 y$=MID$(Button$(x),2*y,1)
266 -24 xx=VAL x$
267 -24 yy=VAL y$
268 -24 IF Flipped(xx,yy,AB)=0 THEN
269 -24 Display(xx,yy,AB)=Display(xx,yy,AB)+Richtung
270 -24 IF Display(xx,yy,AB)>12 THEN Display(xx,yy,AB)=1
271 -24 IF Display(xx,yy,AB)<1 THEN Display(xx,yy,AB)=12
272 -24 Flipped(xx,yy,AB)=1
273 -24 END IF
274 -24 NEXT y

```



Listing. »Gehirntraining« mit »18BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```

275 DD x%=MID$(Button$(x),5,1)
276 MT y%=MID$(Button$(x),6,1)
277 FB xx=VAL(x%)
278 FF yy=VAL(y%)
279 JG Display(xx,yy,3-AB)=Display(xx,yy,3-AB)-Richtung
280 CG IF Display(xx,yy,3-AB)>12 THEN Display(xx,yy,3-AB)=1
281 GT IF Display(xx,yy,3-AB)<1 THEN Display(xx,yy,3-AB)=12
282 JN4 END IF
283 'a2 NEXT x
284 +G Display(2,2,AB)=Display(2,2,AB)+Richtung
285 2N IF Display(2,2,AB)>12 THEN Display(2,2,AB)=1
286 B4 IF Display(2,2,AB)<1 THEN Display(2,2,AB)=12
287 h8 GOSUB PositionOutput
288 .1 FOR x=1 TO 3
289 .14 FOR y=1 TO 3
290 pP6 Flipped(x,y,AB)=0
291 eA4 NEXT y
292 i82 NEXT x
293 h04 'KTJ K
294 49 REM Routine zur Ausgabe des Screenwechsels von 18BIT:
295 2b PositionChange.
296 1r2 IF Seite=0 THEN
297 r44 IF POINT (207,83)<>2+Button(1,1) THEN PAINT (207,83),
2+Button(1,1),13
298 TW IF POINT (312,83)<>2+Button(2,1) THEN PAINT (312,83),
2+Button(2,1),13
299 h4 IF POINT (207,136)<>2+Button(3,1) THEN PAINT (207,136),
2+Button(3,1),13
300 0t IF POINT (312,136)<>2+Button(4,1) THEN PAINT (312,136),
2+Button(4,1),13
301 w4 FOR x=1 TO 3
302 M06 x2=x-1
303 ew FOR y=1 TO 3
304 3b8 IF Display(x,y,1)<>Display(4-x,y,2) THEN
305 pA Display=Display(4-x,y,2)
306 v8 Col=13
307 X0 GOSUB Pointer
308 zH Display=Display(x,y,1)
309 JH Col=15
310 aP GOSUB Pointer
311 xq8 END IF
312 zv6 NEXT y
313 yT4 NEXT x
314 P02 ELSE
315 M54 IF POINT (207,83)<>2+Button(2,2) THEN PAINT (207,83),
2+Button(2,2),13
316 x1 IF POINT (312,83)<>2+Button(1,2) THEN PAINT (312,83),
2+Button(1,2),13
317 v IF POINT (207,136)<>2+Button(4,2) THEN PAINT (207,136),
2+Button(4,2),13
318 f IF POINT (312,136)<>2+Button(3,2) THEN PAINT (312,136),
2+Button(3,2),13
319 a7 FOR x=3 TO 1 STEP -1
320 NR6 x2=3-x
321 wE FOR y=1 TO 3
322 Jw8 IF Display(x,y,2)<>Display(4-x,y,1) THEN
323 A Display=Display(4-x,y,1)
324 G0 Col=13
325 wR GOSUB Pointer
326 Display=Display(x,y,2)
327 v2 Col=15
328 u GOSUB Pointer
329 v88 END IF
330 n6 NEXT y
331 .44 NEXT x
332 iR2 END IF
333 J8 ON Seite+1 GOSUB Blue,Green
334 kM0 RETURN
335 +q REM Routine zum Ausgeben der ersten Zeigerpositionen:
336 Bf PositionPaint:
337 x12 Col=15
338 hX IF Seite=0 THEN
339 B44 FOR x=1 TO 3
340 y06 x2=x-1
341 .1 FOR y=1 TO 3
342 kp8 Display=Display(x,y,1)
343 7y GOSUB Pointer
344 v16 NEXT y
345 Uu4 NEXT x
346 Bu2 ELSE
347 v44 FOR x=3 TO 1 STEP -1
348 pt6 x2=3-x
349 Og FOR y=1 TO 3

```



```

350 h08 Display=Display(x,y,2)
351 P6 GOSUB Pointer
352 896 NEXT y
353 e74 NEXT x
354 ex2 END IF
355 5h0 RETURN
356 AO REM Routine zum Ausgeben der geänderten Zeigerpositionen:
357 5u PositionOutput:
358 1r2 IF Seite=0 THEN
359 .04 x=x-1
360 k8 x=x-1
361 es FOR y=1 TO 3
362 X88 IF Display(x,y,1)<>OldDisplay(x,y,1) THEN
363 QQA Display=OldDisplay(x,y,1)
364 K4 Col=13
365 TK GOSUB Pointer
366 JC IF Sound1=1 THEN GOSUB Sound1
367 wE Display=Display(x,y,1)
368 Col=15
369 X0 GOSUB Pointer
370 .08 END IF
371 w6 NEXT y
372 v4 NEXT x
373 2 ELSE
374 Jw4 FOR x=3 TO 1 STEP -1
375 G06 x2=3-x
376 7 FOR y=1 TO 3
377 v08 IF Display(x,y,2)<>OldDisplay(x,y,2) THEN
378 A44 Display=OldDisplay(x,y,2)
379 2 Col=13
380 i7 GOSUB Pointer
381 f IF Sound1=1 THEN GOSUB Sound1
382 D6 Display=Display(x,y,2)
383 C Col=15
384 m GOSUB Pointer
385 q28 END IF
386 Fw6 NEXT y
387 A74 NEXT x
388 J52 END IF
389 dFO RETURN
390 VP REM Anschalten des Colour Cycling:
391 1M CyclingOn1:
392 uk2 MENU 3,4,1
393 xi MENU 3,3,2
394 G4 Cycle=1
395 ON Seite+1 GOSUB Blue,Green
396 kM0 RETURN
397 EC REM Abschalten des Colour Cycling:
398 ee CyclingOff1:
399 xm2 MENU 3,3,1
400 8u MENU 3,4,2
401 kK Cycle=0
402 qF ON Seite+1 GOSUB Blue,Green
403 TO RETURN
404 REM Darstellung der Grafik von 18BIT mit Hilfe einiger Unte
405 raphen:
406 ex2 GOSUB ST0F
407 TE GOSUB ST0F
408 uu GOSUB ColourDefining
409 d4 GOSUB Blue
410 vF PAINT (0,0),12
411 fx GOSUB Circles
412 5y GOSUB Frame
413 JX PAINT (0,0),0
414 r9 GOSUB Displays
415 n0 GOSUB Buttons
416 hv GOSUB FlipGadget
417 9V GOSUB ArrowGadgets
418 p6 MOUSE ON
419 aP MENU ON
420 8k0 RETURN
421 AT REM Routine zum Darstellen der Kreise fuer den Colour Cycle
422 BY Circles:
423 932 FOR r=217 TO 168 STEP -7
424 k4 CIRCLE (260,109),r,r/7-20
425 Bf PAINT (260,109),r/7-20
426 b02 NEXT r
427 KX FOR r=161 TO 105 STEP -7
428 2K6 CIRCLE (260,109),r,r/7-11
429 G6 PAINT (260,109),r/7-11
430 B42 NEXT r

```




```

431 vH FOR r=98 TO 42 STEP -7
432 xw4 CIRCLE (260,109),r,r/7-2
433 x1 PAINT (260,109),r/7-2
434 j82 NEXT r
435 N40 RETURN
436 y6 REM Routine zum Darstellen des Kanzens von 18BIT
437 AM Frame
438 8P2 LINE (143,20)-(367,20),0
439 FC RESTORE FrameData
440 xN FOR x=1 TO 7
441 E94 READ a1,a2
442 5K LINE -(a1,a2),0
443 422 NEXT x
444 W80 RETURN
445 W9 REM Koordinaten fuer den Rahmen
446 x5 FrameData:
447 7g2 DATA 460,60,460,158,367,198,143,198,60,158,60,60,143,20
448 5t0 REM Routine zum Darstellen des Gadgets zum Seitenwechseln:
449 am FlipGadget:
450 ex2 LINE (467,89)-(508,127),14,b
451 JK LINE (509,127)-(509,89),14
452 n4 LINE (466,90)-(507,128),13,bf
453 w0 COLOR 0,13
454 Rf LOCATE 13,60
455 PY PRINT "FLIP"
456 lK0 RETURN
457 6x REM Routine zum Darstellen der Aktivierungsknoepfe:
458 e0 Buttons:
459 D92 RESTORE ButtonData
460 OA FOR x = 1 TO 4
461 YT4 READ a1,a2
462 7H CIRCLE (a1,a2),17,13,... 36
463 o8 PAINT (a1,a2),2,13
464 2z2 NEXT
465 rT0 RETURN
466 BD REM Koordinaten fuer die Button-ellipse
467 wv ButtonData:
468 vm2 DATA 207,83,312,83,207,136,312,136
469 kJ0 REM Routine zum Darstellen der Displays:
470 i1 Displays:
471 Og2 COLOR 0,1
472 H4 FOR x=1 TO 3
473 Og4 FOR y=1 TO 3
474 hD6 xKoord=154+(x-1)*106
475 Ua yKoord=31+(y-1)*53
476 Km LINE (xKoord,yKoord)-STEP(42,13)
477 PD LINE -STEP(0,24)
478 m1 LINE -STEP(-42,13)
479 w8 LINE -STEP(-42,-13)
480 ep LINE -STEP(0,-24)
481 zW LINE -(xKoord,yKoord)
482 rX LINE (xKoord+10,yKoord+3)-STEP(-20,0)
483 CW PAINT (xKoord,yKoord+2),15,0
484 EC PAINT (xKoord,yKoord+30),13,0
485 g7 FOR a = 1 TO 12
486 tr8 LINE (xKoord,yKoord+25)-(xpos(a)+(x-1)*106,ypos(a)+(y-1)*53)
487 2A6 NEXT a
488 pL4 NEXT y
489 oJ2 NEXT x
490 Gs0 RETURN
491 PV REM Koordinaten fuer die Abschnittslinien der Displays:
492 9x SectorData:
493 WP2 DATA 184,34,184,40,196,50,196,62,184,72,164,78
494 3o DATA 144,78,124,72,112,62,112,50,124,40,144,34
495 FX0 REM Koordinaten fuer die Ecken der Displays
496 DJ SectorData2
497 J12 DATA 154,31,196,44,196,68,154,81,112,68,112,44
498 7b0 REM Routine zum Zeichnen der 8 Gadgets zum Verändern der Zeigerposition
499 l8 ArrowGadgets
500 d62 RESTORE ArrowData
501 av COLOR 14,0
502 aL FOR b=1 TO 2
503 w04 REM Arrow #1:
504 gU6 READ kx,ky
505 in LINE (kx,ky)-STEP(58,7)
506 OG LINE -STEP(0,-15)
507 KR LINE -(kx,ky)
508 x9 PAINT (kx+5,ky),13,14
509 5A4 REM Arrow #2
510 ma6 READ kx,ky
511 W6 LINE (kx,ky)-STEP(-58,7)

```



```

512 G4 LINE -STEP(0,-15)
513 qX LINE (kx,ky)
514 FJ PAINT (kx-5,ky),13,14
515 EK4 REM Arrow #3
516 ag6 READ kx,ky
517 k5 LINE (kx,ky)-STEP(17,25)
518 VC LINE -STEP(-35,0)
519 wd LINE -(kx,ky)
520 Hm LINE (kx,ky+1)-STEP(16,24)
521 kd LINE (kx,ky+1)-STEP(-17,24)
522 8F PAINT (kx,ky+5),13,14
523 PW4 REM Arrow #4
524 Oc6 READ kx,ky
525 H6 LINE (kx,ky)-STEP(17,-25)
526 dK LINE -STEP(-35,0)
527 41 LINE -(kx,ky)
528 hb LINE (kx,ky-1)-STEP(16,-24)
529 K4 LINE (kx,ky-1)-STEP(-17,-24)
530 OX PAINT (kx,ky-5),13,14
531 v2 NEXT b
532 wT0 RETURN
533 h1 REM Koordinaten fuer die Arrow Gadgets:
534 DP ArrowData:
535 kU2 DATA 367,10,143,10,483,158,483,60
536 Hf DATA 367,208,143,208,37,158,37,60
537 ka0 REM Erzeugung von wirklich zufaelligen Werten fuer die Displays.
538 Or New1:
539 s12 RANDOMIZE TIMER
540 MJ FOR x=1 TO 3
541 m4 FOR y=1 TO 3
542 Y16 FOR z=1 TO 2
543 hJ8 OldDisplay(x,y,z)=Display(x,y,z)
544 z5 'splay x,,z =INT z*12**1
545 mJ6 NEXT z
546 lH4 NEXT y
547 kF2 NEXT x
548 e6 FOR x=1 TO 3 STEP 2
549 j94 FOR y=1 TO 3 STEP 2
550 Cl6 Display(x,y,2)=12-Display(x,y,1)
551 R8 IF Display(x,y,2)=0 THEN Display(x,y,2)=12
552 v4 NEXT y
553 q12 NEXT x
554 d8 GOSUB PositionOutput
555 Jv0 RETURN
556 Qn REM Routine zum Abspeichern des momentanen Spielzustandes unter einem eingegebenen Namen:
557 GX SaveAs1.
558 ef2 WINDOW 2,"Spielstand Abspeichern/RETURN für Cancel",(0,0)-(511,8),2,1
559 o6 COLOR 0,1
560 a1 LOCATE 1,1
561 C9 LINE INPUT "Filename: ",Filename$
562 Qw WINDOW CLOSE 2
563 Ov IF Filename$ <> "" THEN OldName$=Filename$: GOSUB Save1
564 lE IF Filename$="" THEN Filename$=OldName$
565 T50 RETURN
566 tP REM Routine zum Abspeichern des momentanen Spielzustandes unter dem alten Dateinamen.
567 XE Save1
568 9F2 IF Filename$="" THEN
569 k54 BEEP: GOSUB SaveAs1
570 nW2 STOP
571 x14 OPEN Filename$ FOR OUTPUT AS 1
572 tP6 FOR x=1 TO 3
573 O18 FOR y=1 TO 3
574 4HA FOR z=1 TO 2
575 7OC PRINT #1,Display(x,y,z)
576 HoA NEXT z
577 z8 NEXT y
578 Tz6 NEXT x
579 q2 FOR x=1 TO 4
580 v18 FOR y=1 TO 2
581 va PRINT #1,Button(x,y)
582 v08 NEXT y
583 tp6 NEXT x
584 i4 PRINT #1,Seite
585 Pz4 CLOSE 1

```

Listing. »Gehirntraining« mit »18BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```

586 OH4 END IF
587 PRO RETURN
588 DX REM Routine zum Abspeichern des momentanen Spielstandes unter
    einem voreingestellten Namen:
589 ZM Quicksave1
590 G42 Filename$="16BIT quick"
591 Kh GOSUB Save1
592 yE Filename$=Oldname$
593 XO RETURN
594 JP REM Routine zum Laden eines abgespeicherten Spielstandes unter
    einem eingegebenen Namen.
595 OS Load1:
596 B02 WINDOW 2,"Spielstand Laden/RETURN für Cancel",(0,0)-(511,
    8),2,1
597 Q1 COLOR 0,1
598 CN LOCATE 1,1
599 o1 LINE INPUT "Filename: ";Filename$
600 s1 WINDOW CLOSE 2
601 j1 IF Filename$<>" " THEN
602 Th4 Oldname$=Filename$
603 Jq File=1
604 vS GOSUB Quickload1
605 Jp File=0
606 N62 ELSE
607 D14 Filename$=Oldname$
608 x62 END IF
609 B40 RETURN
610 I3 REM Laden des mit Quicksave gespeicherten Spielstandes:
611 Mu Quickload1
612 K42 Col=13
613 Sy IF Seite=0 THEN
614 Zv4 FOR x=1 TO 3
615 PR6 x2=x-1
616 h2 FOR y=1 TO 3
617 yC8 Display=Display(x,y,1)
618 fP GOSUB Pointer
619 wS6 NEXT y
620 /Q4 NEXT x
621 eL2 ELSE
622 J44 FOR x=3 TO 1 STEP -1
623 Ok6 x2=3-x
624 p7 FOR y=1 TO 3
625 HR8 Display=Display(x,y,2)
626 gX GOSUB Pointer
627 -a6 NEXT y
628 JY4 NEXT x
629 Sy2 END IF
630 sW IF File=1 THEN Filename1$=Filename$ ELSE Filename1$="16BIT
    Quick"
631 Jo OPEN Filename1$ FOR INPUT AS 1
632 rD4 FOR x=1 TO 3
633 yG6 FOR y=1 TO 3
634 2F8 FOR z=1 TO 2
635 goA INPUT #1,Display(x,y,z)
636 Fu8 NEXT z
637 Rk6 NEXT y
638 D14 NEXT x
639 io FOR x=1 TO 4
640 2J6 FOR y=1 TO 2
641 HCB INPUT #1,Button(x,y)
642 Jp6 NEXT y
643 In4 NEXT x
644 Et INPUT #1,Seite
645 2x2 CLOSE 1
646 fV IF Seite=0 THEN
647 R14 IF POINT (207,83)<>2+Button(1,1) THEN PAINT (207,83),
    2+Button(1,1),13
648 7A IF POINT (312,83)<>2+Button(2,1) THEN PAINT (312,83),
    2+Button(2,1),13
649 J5 IF POINT (207,136)<>2+Button(3,1) THEN PAINT (207,136
    ),2+Button(3,1),13
650 YV IF POINT (312,136)<>2+Button(3,1) THEN PAINT (312,136
    ),2+Button(4,1),13
651 6p2 ELSE
652 nt4 IF POINT (207,83)<>2+Button(2,2) THEN PAINT (207,83),
    2+Button(2,2),13
653 B9 IF POINT (312,83)<>2+Button(1,2) THEN PAINT (312,83),
    2+Button(1,2),13
654 LM IF POINT (207,136)<>2+Button(4,2) THEN PAINT (207,136
    ),2+Button(4,2),13
655 g5 IF POINT (312,136)<>2+Button(3,2) THEN PAINT (312,136
    ),2+Button(3,2),13
656 WP2 END IF
657 sE GOSUB PositionPoint
658 J4 ON Seite+1 GOSUB Blue,Green
659 2B0 RETURN
660 f1 REM Ein versuchter Abbruch durch Break wird unterbunden:
661 s3 BreakTrapping:
662 eL2 BEEP
663 JFO RETURN
664 8F REM Routine zum Erzeugen eines Sounds beim Verstellen der D
    isk sys
665 W3 Sound1
666 x22 OK ai 4 TO 1 STEP -1
667 x44 SOUND 100,,1,ai*ai*16-1
668 -X FOR xi=1 TO 20
669 226 SOUND WAIT
670 f44 NEXT xi
671 ed SOUND RESUME
672 qF2 NEXT ai
673 sp0 RETURN
674 JY REM Anschalten des Sounds:
675 fF SoundOn1:
676 4A2 MENU 3,2,1
677 6 MENU 3,1,2
678 A Sound1=1
679 v0 RETURN
680 t2 REM Abschalten des Sounds.
681 7- SoundOff1:
682 2B2 MENU 3,1,1
683 A MENU 3,2,2
684 -B Sound1=0
685 10 RETURN
686 40 REM Start einer Sequenz:
687 s SeqStart1:
688 1,2 Seq=1
689 W2 WINDOW 2,"Sequenz Abspeichern/RETURN für Cancel",(0,0)-(5
    11,8),2,1
690 vD COLOR 0,1
691 ns LOCATE 1,1
692 sA LINE INPUT "Sequence-/Filename ";Seqname$
693 43 WINDOW CLOSE 2
694 c IF Seqname$="" THEN
695 24 Seq=0
696 f2 ELSE
697 2B4 OPEN Seqname$ FOR OUTPUT AS 2
698 52 END IF
699 4F0 RETURN
700 r REM Stop einer Sequenz:
701 W SeqStop1.
702 262 Seq=0
703 Jx CLOSE 2
704 K0 RETURN
705 22 REM Laden und Zeigen einer Sequenz:
706 Q SeqShow1:
707 1,2 WINDOW 2,"Sequenz Laden/RETURN für Cancel",(0,0)-(511,8),
    2,1
708 vV COLOR 0,1
709 2A LOCATE 1,1
710 2c LINE INPUT "Sequence-/Filename: ";Seqname$
711 f1 WINDOW CLOSE 2
712 H IF Seqname$<>" " THEN
713 24 IF Cycle=1 THEN Cycle1=1 : Cycle=0
714 35 Seite2=Seite
715 52 OPEN Seqname$ FOR INPUT AS 2
716 16 WHILE NOT EOF(2)
717 4a3 FOR x=1 TO 4
718 2A FOR y=1 TO 2
719 oXG INPUT #2,Button(x,y)
720 5A NEXT y
721 38 NEXT x
722 2c INPUT #2,Seite1
723 a INPUT #2,Hebel
724 -c INPUT #2,Richtung
725 o IF Seite1=Seite THEN Seite=1-Seite
726 v0 GOSUB ScreenChange
727 -X GOSUB DisplayPosition
728 2a6 WEND
729 444 CLOSE 2
730 22 IF Cycle1=1 THEN Cycle=1
731 J Cycle1=0
732 42 IF Seite<>Seite2 THEN GOSUB ScreenChange
733 e2 END IF
734 Co0 RETURN
735 J REM Routine zum Abspeichern der einzelnen Sequenzenabschnit
    te.

```



```

736 vx Sequence.
737 ty2 FOR x=1 TO 4
738 ct4 FOR y=1 TO 2
739 xD6 PRINT #2,Button(x,y)
740 tP4 NEXT y
741 sN2 NEXT x
742 tJ PRINT #2,Seite
743 a PRINT #2,Hebel
744 tI PRINT #2,Richtung
745 Ns0 RETURN
746 Bm REM Unterprogramm zur Zeichnen der Zeiger auf dem Display
747 QI Pointer:
748 Xe2 COLOR Col,0
749 XM xKoord=134+x2*106
750 Sf yKoord=56+y-1*53
751 YW IF Display=12 THEN
752 RT4 xk1=xpos(12)+x2*106 yk1=ypos(12)+y-1*53
753 ne xk2=xpos(1)+x2*106 yk2=ypos(1)+(y-1)*53
754 LJ2 ELSE
755 og4 xk1=xpos(Display)+x2*106 yk1=ypos(Display)+(y-1)*53
756 pt xk2=xpos(Display+1)+x2*106 yk2=ypos(Display+1)+(y-1)*53
757 Q22 END IF
758 Sg AREA (xKoord,yKoord)

```

```

759 sm AREA (xk1,yk1)
760 pE IF Display=2=INT(Display/2) AND Display<>12 THEN
761 tP4 AREA(xpos2(Display,2)+x2*106,ypos2(Display/2+1)+(y-1)*53)
762 F72 END IF
763 tI AREA (xk2,yk2)
764 sm AREA (xKoord,yKoord)
765 qw AREAfill
766 qk COLOUR 0,1
767 jP LINE (xKoord,yKoord)-(xk1,yk1)
768 oM IF Display/2=INT(Display/2) AND Display<>12 THEN
769 tB4 LINE -(xpos2(Display,2)+x2*106,ypos2(Display/2+1)+(y-1)*53)
770 MF2 END IF
771 v2 LINE -(xk2,yk2)
772 tD LINE -(xKoord,yKoord)
773 jRO RETURN
774 jZ ENDE
(C) 1989 MBT

```

Listing. Gehirntraining mit »18 BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Schluß).

SPIEL, SPASS SPANNUNG

Spielen macht Spaß.
Vor allem, wenn ein Programm
von der Bedienung, Geschwindigkeit
und Grafik her absolute
Spitze ist: Wie unser »Buh-Au-Construction-Set«.

Eimerkette« oder »Chain Reaction« (Ausgabe 11/87, Seite 66), unter diesen Namen wurden Listings veröffentlicht, die die Grundlage für unser Programm des Monats, das »Buh-Au-Construction-Set« (kurz »BACS«) bilden. Das heißt jedoch nicht, daß dieses Spiel nur eine reine Umsetzung der anderen Listings ist, vielmehr wurde nur das Spielprinzip übernommen. Falls Sie »Eimerkette« nicht kennen, eine kurze Spielbeschreibung: Ziel des Spiels ist es, den Gegner von einem schachbrettartig eingeteilten Spielfeld zu verdrängen. Zu diesem Zweck kann jeder der Spieler Felder belegen. Es darf jedoch pro Spielzug nur jeweils ein Feld belegt werden. Gesetzt werden darf nur auf noch nicht belegte oder schon mit der eigenen Farbe besetzte Felder. Im letzteren Fall erhöht sich der Feldinhalt um eins. Da ein Feld aber nicht unendlich viel aufnehmen kann, »schwappt« der Inhalt über, sobald der Feld-Index gleich der Anzahl der horizontalen und vertikalen Nachbarfelder ist (also maximal vier). Dabei wird jedem der benachbarten Felder die Farbe des »übergelaufenen« Feldes zugeordnet und um eins erhöht, was natürlich einen erneuten »Überlauf« zur Folge haben kann. So kann man durch geschicktes Erhöhen der Felder ganze Ketten aufbauen. Soweit zum Spielprinzip von »Eimerkette«. Wie bereits erwähnt, ist unser Spiel eine Weiterentwicklung. Bei dem Spiel »Buh-Au-Construction-Set« sind noch einige Elemente hinzugekommen:

- Das Spielfeld ist maximal 15 x 15 Felder groß (im Gegensatz zu 7 x 7 bei Eimerkette).
- Es können eigene Spielfelder entworfen werden, so daß man nicht nur auf den mit der Zeit langweiligen, rechteckigen Spielfeldern spielen muß. Die-

se Felder können auf Diskette abgespeichert werden.

- Bis zu drei Spieler können gleichzeitig gegeneinander antreten, was den Spielreiz deutlich erhöht. Bei nur einem Spieler übernimmt der Amiga die Rolle des Gegners.

- Die Bedenkzeit für einen Zug wurde begrenzt. Die maximale Zeit kann von 5 bis 30 Sekunden in 5-Sekunden-Schritten variiert werden.

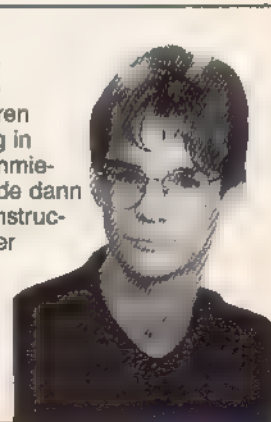
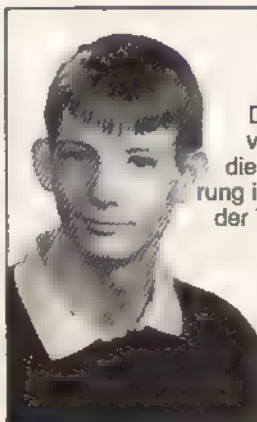
- Nach Spielende werden für jeden Spieler Punkte berechnet und in einer High-Score-Liste auf Disk gespeichert. Für die Verlierer wird eine »Low-Score-Liste« geführt, die ebenfalls gespeichert wird.

- Das ganze Programm ist mausgesteuert. Die Tastatur wird nur zur Eingabe der Spielernamen, beim Speichern und zum Aktivieren einiger Sonderfunktionen wie »Titelaufbau aus/sein«, »Pause« und »Spiel abbrechen« benötigt.

Diese Erweiterungen bewegten die Autoren dazu, den Namen des Spiels zu ändern. Der Name »Buh-Au-Construction-Set« hat keinen tieferen Sinn und soll nicht zum Nachdenken verleiten. Den

MARKUS BONER, KLAUS SCHRECK

Die »Computerlaufbahn« der zwei Programmautoren verlief ähnlich: Ende 1984 fanden sie den Einstieg in die Computerwelt über den C 64. Nach der Programmierung in Basic folgte Assembler. Vor ca. 1,5 Jahren wurde dann der Traumcomputer Amiga angeschafft. »Buh-Au-Construction-Set« ist das erste größere Projekt. Wegen der noch fehlenden Kenntnis des 68000-Assemblers wurde es vernünftigerweise in Basic programmiert. Das »Buh-Au-Construction-Set« wurde Programm des Monats in der AMIGA-Ausgabe 5/89. Wir wünschen den beiden noch weitere erfolgreiche Projekte.



Autoren gefiel aber die Abkürzung so gut. Auf der Diskette müssen sich folgende Dateien befinden:

- Buh-Au_CS
- intuition.bmap
- exec.bmap
- graphics.bmap
- dos.bmap

Falls Sie die „bmap“-Dateien nicht auf Ihrer Extras-Diskette im Verzeichnis »BasicDemos« finden, müssen Sie sie selbst generieren. Eine genaue Anleitung, wie dies funktioniert, finden Sie in der Ausgabe 1/89 des AMIGA-Magazins auf Seite 147.

Da das Programm sehr »speicherfressend« ist, sollten Sie darauf achten, daß beim Aufruf noch mindestens 380 KByte Speicher frei sind. Um das Programm zu starten, verwenden Sie das Ladeprogramm (siehe Listing 1). Beendet werden sollte das Programm nur über die eingebaute »QUIT« Funktion.

Nach dem Starten des Programms werden zunächst die .bmap-Dateien und das Inhaltsverzeichnis der Diskette im internen Laufwerk geladen. Anschließend sucht das Programm nach den High-

- Spielfeld ganz ausfüllen
- Tester.

Dieser Menüpunkt muß aktiviert werden, bevor gespielt werden kann. Das Spielfeld wird nach Aufruf dieses Punktes auf seine Spielbarkeit hin überprüft. Spielbare Felder dürfen keine Einzelfelder enthalten, das heißt, jedes Feld muß mindestens zwei Nachbarfelder haben. Weiterhin muß das Spielfeld in einem durchgehen, das heißt, es darf nur ein abgeschlossenes Spielfeld vorhanden sein. Der Tester meldet solche unkorrekten Spielfelder entweder mit »Einfeld vorhanden« oder mit »Kein korrektes Spielfeld«. Zugelassene Felder werden mit »Spielfeld ist zugelassen« quittiert.

- Menü

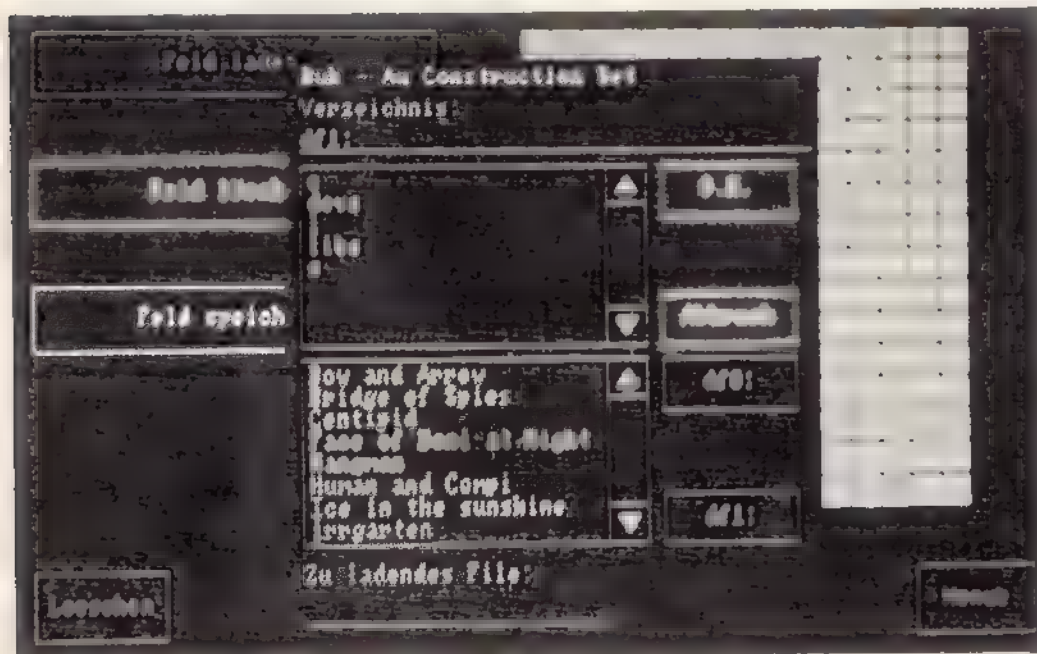
Rücksprung ins Hauptmenü.

Für sämtliche Diskettenoperationen wird ein Filerequester, der selbständig ein angeschlossenes Laufwerk DF1. erkennt, verwendet. Dessen Bedienung wird kurz erläutert: Oben wird in Fettdruck der Diskettenname angegeben, darunter das aktuelle Verzeichnis. Im oberen der beiden Kästen sind die Unterverzeichnisse aufgelistet, in die durch einfaches Anklicken verzweigt werden kann (mit gedrückter Maustaste kann aufwärts oder abwärts »gefahren« werden). Durch Anklicken der kleinen Pfeile kann der Anzeigebereich verschoben werden. Das gleiche gilt für den »File-Kasten« darunter. Jeder Lade-, Speicher- oder Löschvorgang muß durch Anklicken von »O.K.« bestätigt werden. Zur Eingabe des Savefile-Namens muß unter »Zu speicherndes File« geklickt werden, worauf der Cursor erscheint. Nachdem ein Spielfeld entworfen oder geladen wurde, müssen noch die Spielstellungen vorgenommen werden. Nach Anklicken der entsprechenden Box im Hauptmenü können folgende Einstellungen vorgenommen werden:

- Die maximale Denkzeit für einen Zug (einfach in der oberen Reihe die gewünschte Zeit anklicken).
- Die Anzahl der Spieler. Die Spielernamen werden nach der Auswahl der Spieleranzahl automatisch abgefragt. Durch Anklicken von »Menü« gelangen Sie wieder in das Hauptmenü. Jetzt kann das Spiel gestartet werden.

Sollte irgendeine Operation nicht korrekt ausgeführt worden sein, so meldet der Computer, was noch gemacht werden muß.

Während des Spiels ist der Bildschirm wie folgt aufgeteilt: Die rechte Hälfte beinhaltet das Spielfeld. Auf der linken Seite wird angezeigt, wer gerade am Zug ist. Unter dieser Anzeige sind die Spielernamen aufgeführt. Hinter diesen wird die vom Spieler jeweils schon in Anspruch genommene Denkzeit angegeben. In der Mitte oben wird angezeigt, wieviel Zeit dem Spieler noch für seinen Zug verbleibt. Die letzten drei Sekunden werden durch Pieptöne begleitet. Durch Druck auf die linke Maustaste wird das Spiel und die Zeit des ersten Spielers gestartet. Um auf ein Feld zu setzen, wird einfach mit der Maus das entsprechende Feld angeklickt. Falls das Feld nicht besetzt werden darf (weil es schon ein anderer besetzt hat), wird dies durch einen Pieps mitgeteilt. Durch Druck auf <F10> läßt sich das Spiel in den Pausenmodus bringen. Zum Weiterspielen einfach die linke Maustaste drücken. Wenn man keine Lust hat, das Spiel zu Ende zu spielen, kann es durch <CTRL Q> abgebrochen werden. Zur Sicherheit wird das Spielfeld ausgeblendet (auch bei Pause) und eine Sicherheitsabfrage durchgeführt. Nach Spielende wird der Gewinner bekanntgegeben und die High- und Low-Score-Liste gezeigt. Falls ein Spieler in eine der beiden Listen aufgenommen wurde, wird diese automatisch auf Diskette gespeichert. Ins Hauptmenü gelangt man durch einen Druck auf die linke Maustaste. Während der Benutzung des Programms kommt es vor, daß der Computer eine Meldung zu machen hat. Sie erscheint rot und zentriert auf dem Bildschirm. Um das Programm fortzusetzen, drücken Sie einfach die linke Maustaste.



Spiel, Spaß und Spannung für viele Stunden bietet das »Buh-Au-Construction-Set«

Scores und generiert eine eigene High-Score-Liste, falls keine auf Disk gefunden werden. Diese Diskettenoperationen dauern natürlich etwas (man hat genug Zeit, zur Kaffeemaschine zu kommen). Danach erscheint das Titelbild, das gleichzeitig als Hauptmenü fungiert und im folgenden auch so bezeichnet wird. Da auch der Titelaufbau recht lange dauert, kann er durch Druck auf <F1> an- und ausgeschaltet werden (die Einstellung wird erst beim nächsten Aufbau des Hauptmenüs berücksichtigt). Nun zu den einzelnen Menüpunkten, die durch einfaches Anklicken der Schalter ausgewählt werden.

■ Spielfeld konstruieren

Bevor ein Spiel gestartet werden kann, muß erst ein Spielfeld konstruiert werden. Zu diesem Zweck begibt man sich in das Construction-Set. Nach kurzer Wartezeit ist man im Konstruktionsmenü. Auf der rechten Seite des Bildschirms ist eine 15 x 15 Felder große Matrix, die dem Spielfeld entspricht. Durch Anklicken einzelner Felder können diese entweder gesetzt oder gelöscht werden. Wird die Maustaste gedrückt gehalten, setzt BACS kontinuierlich Punkte.

Die anderen Funktionen haben folgende Bedeutungen.

- Feld laden.
Auf Diskette gespeicherte Felder werden geladen.
- Feld löschen
Auf Diskette gespeicherte Felder löschen
- Feld speichern:
Feld auf Disk abspeichern
- Löschen:
Spielfeld löschen
- Füllen:

Der letzte Hauptmenüpunkt ist »QUIT«. Nach Anklicken der Box wird noch einmal eine Sicherheitsabfrage durchgeführt.

Wenn Sie außer Spielen auch noch das Programmieren im Kopf haben, sollten Sie sich das Programm etwas genauer ansehen. Durch die Benutzung des Betriebssystems ist das Programm komfortabel und schnell. Die Auswahl einer Datei sollte immer über einen Filerequester geschehen, wie BACS es vornimmt. Fehleingaben sind dadurch vermeidbar. Außerdem muß der Benutzer nicht raten, wie die Datei heißt. Überprüfungen, wie BACS sie vornimmt, be spielsweise kein Spielstart ohne vorherige Voreinstellungen, sind beispielhaft. Übrigens finden Sie auf unserer Programmser vice-Diskette 15 fertige Spielfelder. Das heißt viele Stunden Spiel, Spaß und Spannung...

Markus Boner/Klaus Schreck/kn

Programmname:	Lade_Buh-Au_CS
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Bemerkung:	Startprogramm
Programmautor: Markus Boner, Klaus Schreck ----- 1 W80 CLEAR,45000& 2 Xn LOAD"buh-au_cs",x (C) 1989 M&T	

Listing 1. Das Ladeprogramm zum »Buh-Au-Construction-Set« lädt das Hauptprogramm

Programmname:	Buh-Au_CS
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Programmautor: Markus Boner, Klaus Schreck ----- 1 yFO LIBRARY"graphics.library" 2 aM LIBRARY"exec.library" 3 LY LIBRARY"dos.library" 4 CZ LIBRARY"intuition.library" 5 UL DECLARE FUNCTION AllocMem& LIBRARY 6 Oc DECLARE FUNCTION Examine& LIBRARY 7 Jq DECLARE FUNCTION ExNext& LIBRARY 8 f9 DECLARE FUNCTION Lock& LIBRARY 9 qX DECLARE FUNCTION UnLock& LIBRARY 10 BU DECLARE FUNCTION OpenDevice& LIBRARY 11 a6 DECLARE FUNCTION CloseDevice& LIBRARY 12 QD exk=217:info&=AllocMem&(252&,exk) 13 hp a\$="trackdisk.device"+CHR\$(0) 14 Qx exk=OpenDevice&(&SADD(a\$),1&,info&,0).df1%<exk<>0 15 pk IF NOT df1% THEN 16 Uv2 exk=CloseDevice&(info&) 17 D60 END IF 18 Tc RANDOMIZE TIMER 19 Z3 DEF FN Maus=PEEK(12574721&) AND 64 20 ti DEFINT a,e,l,x,y,g,k,h,j,z 21 be DIM abc\$(3),min\$(3),sec\$(3),feld\$(14,14),init\$(14,14),feld farbe\$(14,14),dir\$ 100. 22 kI DIM stapel\$(225,2),wert\$(3),spielernamen\$(3),punkte\$(3),naa en\$(3),file\$ 100 23 Vy DIM hsc\$(13),hsc\$(13),lsc\$(13),lsc\$(13),eintr(100,1) 24 Zt drive\$="":GOSUB LiesDiskName 25 mN rDisk\$ file\$,0 a\$="rDisk\$"+SCORES bos"+CHR\$(26 Qe exk=Lock& SADD(a\$, 2 ScoreFehler%-exk=0 27 7 exk=UnLock&(exk, disk\$="dfo " drive\$=disk\$ 28 od GOSUB LeseDir GOSUB HighScores 29 SH WINDOW CLOSE 1:SCREEN 1,640,255,3,2:WINDOW 2,,,16,1 30 Q9? FarbenLoeschen 31 B3 ON MOUSE GOSUB MouseTest 32 PQ ON TIMER(1) GOSUB ZeitAuswertung 33 dN GOSUB Loeschen:title%=-1:flag%=1:GOSUB Aufbau1 MOUSE LA 34 oS Endlos. 35 In IF komp AND zahl=2 THEN GOSUB Computer 36 gE a\$=INKEY\$ 37 kd IF a\$<>" " THEN 38 DR2 IF a\$=CHR\$(129) AND flag%=1 THEN 39 L34 MOUSE OFF:SOUND 800,2,255,0:title%=NOT title%	

40 Yr	Wertung\$="Titelaufbau "
41 ZH	IF title% THEN Wertung\$=Wertung\$+"ein" ELSE Wertung\$=W ertung\$+"aus"
42 vI	Meldung Wertung\$,2:MOUSE ON
43 SW2	END IF
44 Pn	IF a\$=CHR\$(17) AND flag%=24 THEN
45 J14	TIMER OFF:MOUSE OFF
46 JS	WINDOW 4,,(300,30)-(570,210),0,1
47 cm	GOSUB Sicher:WINDOW CLOSE 4
48 J7	IF ja THEN
49 936	CLS:flag%=1
50 Ma	GOSUB Neustart:initialisierungsRoutine
51 ao	IF komp THEN komp=0:spanz%=1
52 ez	GOSUB Aufbau1
53 SB5	ELSE
54 616	flag%=24:"TIMER ON
55 p14	END IF
56 s6	MOUSE ON
57 h2	END IF
58 S3	IF a\$=CHR\$(136) AND flag%=24 THEN
59 w74	TIMER OFF:MOUSE OFF:WINDOW 4,,(300,30)-(570,210),0,1
60 gq	COLOUR 5,0:LOCATE 7,8:Printe "- P - A - U - S - E - "
61 hI	COLOR 4,0:LOCATE 10,7:Printe "Zum Weiterspielen bitte"
62 PD	LOCATE 12,4:Printe "die linke Maustaste dr"+CHR\$(252)+ "cken."
63 OJ	CALL waitForClick WINDOW CLOSE 4:MOUSE ON:TIMER ON
64 yx2	END IF
65 S1	WHILE .INKEYS<>" " WEND
66 010	END IF
67 la	IF MOUSE 0 = 0 THEN Endlos ELSE Endlos
68 7	MouseTest
69 de	mx=MOUSE 1, my=MOUSE 2
70 ye	IF flag%=1 THEN RESTORE Daten1
71 Ya	IF flag%>2 AND flag%<7 OR flag%=20 THEN RESTORE Daten2
72 pz	IF flag%=11 OR flag%=22 THEN RESTORE Daten4
73 23	IF flag%=13 OR flag%=23 THEN RESTORE Daten5
74 FK	IF flag%=15 OR flag%=24 THEN RESTORE Daten6
75 vI	IF flag%=-3 OR flag%=-2 THEN RESTORE Daten7
76 X	IF flag%
77 ei	loc
78 MZ	READ x1
79 -Q	IF x1=-1 THEN flag%=j:RETURN
80 dp	READ y1,x2,y2,flag%,farbe\$,wort\$
81 H6	IF dx>=x1 AND dx<=x2 AND dy>=y1 AND dy<=y2 THEN
82 x12	MOUSE OFF
83 Wg	IF flag%<20 THEN
84 vJ4	Zeichnen2 x1,y1,x2,y2,farbe\$,wort\$
85 lY	IF flag%<-1 THEN GOSUB ja:RETURN
86 cR	ON flag% GOSUB Aufbau1,Quit,Aufbau2,Loeschen,Checker,F uellen,Out,Out,,Aufbau3,Anklicken2,Spielauswahl,A bbruch,Aufbau4
87 UI	MOUSE ON
88 m0	RETURN
89 213	ELSE
90 A14	ON (flag%-19) GOSUB Anklicken,,Anklicken2,SpielerEinga be,Setzen
91 Yp	MOUSE ON
92 qS	RETURN
93 RK2	END IF
94 S10	END IF
95 WJ	GOTO Neu
96 Cx	Daten1:
97 JG	DATA 60,140,300,172,3,2,Spielfeld konstruieren,340,140,580 172,11,3,Spieleinstellung
98 Sn	DATA 60,184,300,216,15,4,Spiel starten,340,184,580,216,2,5 2,5,1,T,-1
99 H3	Daten2:
100 G2	DATA 300,0,570,180,20,0,,8,200,88,224,4,6,Loeschen,360,200 440,224,5,4,Tester,536,200,608,224,1,3,Menne
101 SX	DATA 8,0,248,24,7,7,Feld laden,8,96,248,120,8,5,Feld spelo hern,176,200,248,224,6,2,Fuellen
102 va	DATA 8,49,248,73,9,4,Feld l5schen,-1
103 VA	Daten3:
104 v	DATA 220,38,300,57,7,4,0,K.,220,86,300,105,8,4,Abbruch,220 110,300,129,9,3,"dfo:"
105 2r	DATA 220,159,300,177,10,3,"df1:"4,38,186,105,1,,186,38,2 10,52,2,,186,91,210,105,3,,4,110,186,177
106 xV	DATA 4,,186,110,210,124,5,,186,163,210,177,6,,4,194,186

Listing 2. Ein Spiel mit »allem Drum und Dran«:
»Buh-Au-Construction-Set«. Bitte mit dem Checksummer
(Seite 143) eingeben (Fortsetzung).


```

248 c3 Gotoxy gx,gy:COLOR zahl+2,2:PRINT CHR$(feld$(x,y)+48)
249 z7 IF feld$(x,y)=init$(x,y) THEN i=0:GOSUB Ueberlauf
250 bp IF spanz%-null=1 THEN GOSUB Endauswertung:RETURN
251 g3 zahl=zahl MOD spanz%+1
252 bc IF wert$(zahl)=-1 THEN zahl=zahl MOD spanz%+1
253 b4 GOSUB NeuerSpieler
254 jp TIMER ON
255 ir1 ELSE
256 g32 SOUND 1000,4,255,0
257 y0 END IF
258 w8 RETURN
259 eb Ueberlauf
260 ag IF x THEN
261 gq2 IF feld$(x-1,y)<>-1 THEN
262 fj4 IF feldfarbe$(x-1,y)<>zahl+2 AND feldfarbe$(x-1,y) T
HEN
263 aj6 wert$((feldfarbe$(x-1,y)-2)=wert$(feldfarbe$(x-1,y)
))-2,-feld$(x-1,y,
wert$(zahl)=wert$(zahl)+feld$(x-1,y
264 t0 END IF
265 d64 feld$(x-1,y)=feld$(x-1,y)+1:feldfarbe$(x-1,y)=zahl+2
266 wf IF (feld$(x-1,y) MOD init$(x-1,y))=0 THEN i=i+1:stapel
%1,1=x-1:stapel$(1,2)=y
267 nj Gotoxy gx-18,gy:PRINT CHR$(feld$(x-1,y)+48)
268 nc END IF
269 ha2 END IF
270 ib0 END IF
271 lq IF x<14 THEN
272 hv2 IF feld$(x-1,y)<>-1 THEN
273 b14 IF feldfarbe$(x-1,y)<>zahl+2 AND feldfarbe$(x-1,y) T
HEN
274 vs6 wert$((feldfarbe$(x-1,y)-2)=wert$(feldfarbe$(x-1,y)
))-2,-feld$(x-1,y,
wert$(zahl)=wert$(zahl)+feld$(x-1,y)
275 v3 END IF
276 oh4 feld$(x-1,y)=feld$(x-1,y)+1:feldfarbe$(x-1,y)=zahl+2
277 r4 IF (feld$(x-1,y) MOD init$(x-1,y))=0 THEN i=i+1:stapel
%1,1=x-1:stapel$(1,2)=y
278 aq Gotoxy gx-18,gy:PRINT CHR$(feld$(x-1,y)+48)
279 af END IF
280 sl2 END IF
281 tmo END IF
282 bs IF y THEN
283 ku2 IF feld$(x,y-1)<>-1 THEN
284 bv4 IF feldfarbe$(x,y-1)<>zahl+2 AND feldfarbe$(x,y-1) T
HEN
285 is6 wert$((feldfarbe$(x,y-1)-2)=wert$(feldfarbe$(x,y-1)
))-2,-feld$(x,y-1,
wert$(zahl)=wert$(zahl)+feld$(x,y-1)
286 xs END IF
287 z54 feld$(x,y-1)=feld$(x,y-1)+1:feldfarbe$(x,y-1)=zahl+2
288 es IF (feld$(x,y-1) MOD init$(x,y-1))=0 THEN i=i+1:stapel
%1,1=x:stapel$(1,2)=y-1
289 di Gotoxy gx,gy-12:PRINT CHR$(feld$(x,y-1)+48)
290 pb END IF
291 dw2 END IF
292 ex0 END IF
293 ms IF y<14 THEN
294 hv2 IF feld$(x,y+1)<>-1 THEN
295 gm4 IF feldfarbe$(x,y+1)<>zahl+2 AND feldfarbe$(x,y+1) T
HEN
296 bf6 wert$((feldfarbe$(x,y+1)-2)=wert$(feldfarbe$(x,y+1)
))-2,-feld$(x,y+1,
wert$(zahl)=wert$(zahl)+feld$(x,y+1)
297 yr END IF
298 xd4 feld$(x,y+1)=feld$(x,y+1)+1:feldfarbe$(x,y+1)=zahl+2
299 xf IF (feld$(x,y+1) MOD init$(x,y+1))=0 THEN i=i+1:stapel
%1,1=x:stapel$(1,2)=y+1
300 sr Gotoxy gx,gy+12:PRINT CHR$(feld$(x,y+1)+48)
301 mu END IF
302 oh2 END IF
303 ni0 END IF
304 lm feld$(x,y)=feld$(x,y)+init$(x,y)
305 oh Gotoxy gx,gy
306 j9 IF feld$(x,y)=0 THEN PRINT " ":feldfarbe$(x,y)=0 ELSE PRIN
T CHR$(feld$(x,y)+48)
307 mh =0
308 z8 FOR k=1 TO spanz%
309 am2 IF wert$(k)=0 THEN null=null+1:wert$(k)=NOT wert$(k):IF
spanz%-null<>1 THEN Meldung spielernamen$(k)+" ist ausge
schieden",k+2 ELSE k=spanz%:j=1
310 qao NEXT
311 d7 IF j THEN RETURN
312 wh IF i THEN x=stapel$(i,1):y=stapel$(i,2):gx=305+(x*5+(10*x
))gy=39+(y*8+(4*y)):i=i-1:GOTO Ueberlauf
313 p1 RETURN

```

```

314 gc Computer-
315 to MOUSE OFF:DIM kont(255,2) i=0
316 o3 FOR y=0 TO 14
317 y12 FOR x=0 TO 14
318 s93 IF feldfarbe$(x,y)=zahl+2 AND feld$(x,y)=init$(x,y)-1 T
HEN i=i+1:kon$(i,1)=x:kon$(i,2)=y
319 z72 NEXT
320 x0 NEXT
321 lx hprior=0:k=0
322 ec WHILE hz>1 AND k<1
323 z22 prior=0:x=k+1:x=kont(k,1):y=kont(k,2)
324 p IF x>0 THEN
325 a44 IF feldfarbe$(x-1,y)=zahl+1 THEN
326 f16 prior=prior+1
327 al IF feld$(x-1,y)=init$(x-1,y)-1 THEN prior=prior+1
328 z74 END IF
329 z32 END IF
330 h4 IF x<14 THEN
331 a44 IF feldfarbe$(x+1,y)=zahl+1 THEN
332 x66 prior=prior+1
333 z9 IF feld$(x+1,y)=init$(x+1,y)-1 THEN prior=prior+1
334 z24 END IF
335 le2 END IF
336 ph IF y>0 THEN
337 o44 IF feldfarbe$(x,y-1)=zahl+1 THEN
338 v66 prior=prior+1
339 va IF feld$(x,y-1)=init$(x,y-1)-1 THEN prior=prior+1
340 z34 END IF
341 rk2 END IF
342 z7 IF y<14 THEN
343 mo4 IF feldfarbe$(x,y+1)=zahl+1 THEN
344 i36 prior=prior+1
345 p1 IF feld$(x,y+1)=init$(x,y+1)-1 THEN prior=prior+1
346 wp4 END IF
347 xq2 END IF
348 yf IF prior>hprior THEN j=k:hprior=prior
349 zso WEND
350 a5 IF wert$(2)>3 OR hprior<>0 THEN
351 mo2 IF i>0 THEN
352 w04 IF hprior=0 THEN k=INT(RND*1)+1:x=kont(k,1):y=kont(k,2)
ELSE x=kont(j,1):y=kont(j,2)
353 dw2 END IF
354 ex0 END IF
355 lk IF i=0 OR (wert$(2)<4 AND hprior=0) THEN
356 kb2 x=INT(RND*15):y=INT(RND*15)
357 lv WHILE feldfarbe$(x,y)=zahl+1 OR feld$(x,y)=-1
358 md x=INT(RND*15):y=INT(RND*15)
359 f2 ENL
360 fd0 END IF
361 ou ERASE kont:GOSUB Setzen2:MOUSE ON
362 co RETURN
363 ly ZeitAuswertung:
364 d7 IF flag%=24 OR flag%=15 THEN
365 bk1 IF NOT komp OR (zahl=1 AND komp) THEN MOUSE STOP
366 r sec$ zahl=sec$(zahl)+1
367 st IF sec$(zahl)=60 THEN sec$(zahl)=0:min$(zahl)=min$(zahl)+
1
368 fd COLOR zahl+2,0: LOCATE 13+2*zahl,25
369 zy IF min$(zahl)<10 THEN PRINT "0";
370 y3 PRINT RIGHTS$(STR$(min$(zahl)),27-POS(0));". ";
371 o3 IF sec$(zahl)<10 THEN PRINT "0";
372 z8 PRINT RIGHTS$(STR$(sec$(zahl)),30-POS(0))
373 kv hz=hz-1
374 sb IF hz<4 THEN SOUND 800,2,200,3
375 vs IF hz=0 THEN
376 z3 zahl=zahl MOD spanz%+1
377 z4 IF wert$(zahl)=-1 THEN zahl=zahl MOD spanz%+1
378 x SOUND 800,4,200,3:GOSUB NeuerSpieler
379 zw1 END IF
380 yv LOCATE 2,47:PRINT USING"##";hz
381 lg IF NOT komp OR (zahl=1 AND komp, THEN MOUSE ON
382 ec0 END IF
383 qv RETURN
384 lq NeuerSpieler-
385 xk COLOR zahl+2,0
386 gx n=zahl+1
387 gb LOCATE 2,31:Printe "Denkzeit "+CHR$(252)+"brig: ":PRINT US
ING"##";hz

```

Listing 2. Ein Spiel mit «allem Drum und Dran»:
»Buh-Au-Construction-Set«. Bitte mit dem Checksummer
 (Seite 143) eingeben (Fortsetzung).

```

388 9n LOCATE 10,2:Printe spielername$(zahl)+" 1st am Zug."+SPACE
    $(20-LEN(spielername$(zahl)))
389 aF RETURN
390 tp Checker
391 vp IF zeiger THEN
392 fD2 GOSUB Wertung
393 hJ RETURN
394 lB0 END IF
395 9g IF anzahl<4 THEN
396 l22 IF anzahl=0 THEN zeiger=-3 ELSE zeiger=-2
397 nI GOSUB Wertung
398 m0 RETURN
399 hQ0 END IF
400 Up IF einrflD THEN
401 N52 FOR i=1 TO einrflD
402 JK4 x=einr(1,0):y=einr(1,1):GOSUB Kaestchen
403 af2 NEXT
404 SLQ END IF
405 ZS ERASE init DIM init$(14,14)
406 rg gesamt=0 zeiger=0 i=1 j=anzahl einrflD=0
407 Jp FOR yt=0 TO 14
408 fD2 FOR xt=0 TO 14
409 EX4 IF feld$(xt,yt)=0 THEN feld$(xt,yt)=1:GOSUB Checkbegin
    yt=14 xt=1:
410 hm2 NEXT
411 in0 NEXT
412 H9 IF zeiger<>-4 THEN
413 xA2 IF j=0 THEN zeiger=1 ELSE zeiger=-1
414 cv0 END IF
415 lQ GOSUB Loeschen2 GOSUB Wertung
416 4g RETURN
417 mk Checkbegin
418 lS WHILE 1
419 vr i=i+1
420 91 IF init$(xt,yt)=0 THEN
421 l62 IF xt THEN
422 q34 IF feld$(xt-1,yt)>-1 THEN
423 xK6 feld$(xt-1,yt)=1:init$(xt,yt)=init$(xt,yt)+1
424 tU IF init$(xt-1,yt)=0 THEN i=i+1:stapel$(1,1)=xt-1:sta
    pel$(1,2)=yt
425 ng4 END IF
426 on2 END IF
427 NK IF xt<14 THEN
428 q74 IF feld$(xt+1,yt)>-1 THEN
429 yE6 feld$(xt+1,yt)=1:init$(xt,yt)=init$(xt,yt)+1
430 nK IF init$(xt+1,yt)=0 THEN i=i+1:stapel$(1,1)=xt+1:sta
    pel$(1,2)=yt
431 tm4 END IF
432 un2 END IF
433 z0 IF yt THEN
434 vK4 IF feld$(xt,yt-1)>-1 THEN
435 lv6 feld$(xt,yt-1)=1:init$(xt,yt)=init$(xt,yt)+1
436 SE IF init$(xt,yt-1)=0 THEN i=i+1:stapel$(1,1)=xt:stape
    l$(1,2)=yt-1
437 zb4 END IF
438 Ot2 END IF
439 en IF yt<14 THEN
440 p04 IF feld$(xt,yt+1)>-1 THEN
441 t66 feld$(xt,yt+1)=1:init$(xt,yt)=init$(xt,yt)+1
442 E6 IF init$(xt,yt+1)=0 THEN i=i+1:stapel$(1,1)=xt:stape
    l$(1,2)=yt+1
443 5y4 END IF
444 6z2 END IF
445 bB IF init$(xt,yt)=1 THEN
446 bE4 einrflD=einrflD+1:einr(einrflD,0)=xt
447 Hg einr(einrflD,1)=yt zeiger=-4 x3=300+xt*18.x4=316+xt*18
448 MB y3=yt*12 y4=10+yt*12 LINE(x3,y3)-(x4,y4),4,bf
449 B42 END IF
450 Pl j=j-1 gesamt=gesamt+init$(xt,yt)
451 D60 END IF
452 rM xt=stapel$(1,1).yt=stapel$(1,2)
453 kY wEND
454 gI RETURN
455 O4 Loeschen.
456 Gp FOR y=0 TO 14
457 Bo2 FOR x=0 TO 14
458 re4 feld$(x,y)=-1
459 Fg IF flag%=4 THEN GOSUB Kaestchen
460 la2 NEXT
461 wB0 NEXT
462 HL IF flag%=4 THEN
463 Xf2 Neusetzen x1,y1,x2,y2,farbe$,wort$:anzahl=0:zeiger=0
464 Q70 END IF

```

```

465 rT RETURN
466 Tg Fuellen.
467 RO FOR y=0 TO 14
468 qV2 FOR x=0 TO 14:feld$(x,y)=0:GOSUB Kaestchen:NEXT
469 eJ0 NEXT
470 fu Neusetzen x1,y1,x2,y2,farbe$,wort$:anzahl=225:zeiger=0
471 xZ RETURN
472 qP Loeschen2.
473 X6 FOR y=0 TO 14
474 AB2 FOR x=0 TO 14:IF feld$(x,y)=1 THEN feld$(x,y)=0
475 kp NEXT
476 ag0 NEXT
477 3f RETURN
478 xL Wertung:
479 oB RESTORE Fehlermeldung
480 ta j=zeiger
481 NH IF zeiger=1 THEN j=0
482 QR FOR i=1 TO j+5:READ Wertung$:NEXT
483 zN Meldung wertung$,4:Neusetzen x1,y1,x2,y2,farbe$,wort$
484 Am RETURN
485 Zw Fehlermeldung
486 49 DATA Einerfeld vorhanden,Kein Spielfeld vorhanden,Zu klein
    es Spielfeld
487 i1 DATA Kein Korrektes Spielfeld,Spielfeld ist ausgelassen
488 rI Anklicken:
489 va IF POINT(mx,my)=0 OR POINT(mx,my)=1 THEN RETURN
490 fa xh=(mx-300)\18:yh=my\12 x=xh y=yh
491 va Marke.
492 l22 IF feld$(x,y)=-1 THEN
493 X04 feld$(x,y)=0:anzahl=anzahl+1
494 7I3 ELSE
495 wa4 feld$(x,y)=-1:anzahl=anzahl-1
496 wp2 END IF
497 z2 GOSUB Kaestchen
498 GI WHILE x=xh AND y=yh AND FN Maus=0
499 xd4 mx=MOUSE(0):mx=MOUSE(5):my=MOUSE(6)
500 bA xh=(mx-300)\18:yh=my\12
501 p2 IF yh>14 OR xh>14 OR yh<0 OR xh<0 THEN y=yh:x=xh
502 XL2 wEND
503 BT x=xh:y=yh
504 Yx0 IF FN Maus=0 THEN GOTO Marke
505 a4 zeiger=0
506 w6 RETURN
507 Ab Anklicken2.
508 p2 IF hwort$<>" THEN
509 v2 LINE (x7+1,y7+1)-(x8-1,y8-1),0,bf
510 r9 COLOR 2,0:LOCATE 2,3+10*(x7\80):Printe hwort$
511 B40 END IF
512 Ab LINE (x1,y1)-(x2,y2),2,bf:COLOR 0,2
513 lS LOCATE 2,3+10*(x1\80):Printe wort$:COLOR 3,0
514 Gr x7=x1 y7=y1 x8=x2 y8=y2:hwort$=wort$ zeit=5*(x1\80)
515 FH RETURN
516 Dk Sicher.
517 RL Wertung$="Sind Sie 100%ig sicher ??":GOSUB Requester
518 lA RETURN
519 YN Sicher2.
520 me Wertung$="File existiert !! SAVE ??":GOSUB Requester
521 lN RETURN
522 S3 Requester:
523 Zm flag%=3
524 ZE whileMousePressed
525 CQ WINDOW 3,,(196,58)-(428,101),16,1:LINE (0,0)-(232,52),6,b
526 t0 RESTORE Daten7:Zeichnen 2
527 t2 COLOR 4,0:LOCATE 2,3:Printe wertung$.ja=1
528 lt WHILE ja=1
529 tM2 call WaitForClick:mx=MOUSE(0):GOSUB MouseTest
530 zn0 wEND
531 qX WINDOW CLOSE 3
532 wY RETURN
533 oT ja.
534 AS IF flag%=-2 THEN ja=-1 ELSE ja=0
535 zb RETURN
536 nh Out.
537 5B Neusetzen x1,y1,x2,y2,farbe$,wort$
538 JM num=flag%-6 c$=file$(0)
539 RW GOSUB LiesDiskName:wBenchToBack
540 pB IF c$<>file$(0) THEN disk$=drive$:GOSUB LiesDir
541 rC efile=1:edir=0:WINDOW 3,,(165,14)-(475,218),0,1
542 po COLOR 6,0:Gotoxy 4,6:POKE WINDOW(8)+56,2:Printe file$(0)
543 X4 POKE WINDOW(8)+56,0:Gotoxy 4,18:Printe "Verzeichnis: "
544 DX Gotoxy 4,30:Printe disk$:LINE (4,32)-(438,32),6
545 nt FileKasten 0,farbe$:GOSUB DirAnzeige:FileKasten 72,farbe$
546 Sn GOSUB FileAnzeige

```



```

547 CE a$=RIGHT$(wort$,LEN(wort$)-INSTR(wort$," "))
548 uX Gotoxy 4,190 Printe "Zu "+a$+"des File:"
549 zu Gotoxy 4,202 Printe namen$.num)
550 GS LINE (4,205)-(186,205).RESTORE Daten3:Zeichnen 4+df1%:k=0
551 K6 WHILE k<7 OR k>8
552 dQ2 call WaitForClick:mx=MOUSE(0):my=MOUSE(1)
553 4u my=MOUSE(2):k=0:x1=0
554 aR RESTORE Daten3
555 GC WHILE x1<>-1 AND k=0
556 ar4 READ x1,y1,x2,y2,flag%,i,wort$
557 E6 IF x1<>-1 AND mx>x1 AND my>y1 AND mx<x2 AND my<y
2 THEN k=flag%
558 RF2 WEND
559 30 IF k>6 AND k<11+df1% THEN farbe%=i:Zeichnen2 x1,y1,x2,
y2,farbe%,wort$
560 D3 IF x1<>-1 THEN
561 rs4 ON k GOSUB SelDir,DirUp,DirDown,SelFile,FileUp,FileDow
n,Ok,Abbruch,ChDisk,ChDisk,SaveFile
562 Dt2 END IF
563 WK0 WEND
564 S4 RETURN
565 eP ChDisk.
566 ZM IF k=10 AND df1% THEN RETURN
567 Zf Neusetzen x1,y1,x2,y2,farbe%,wort$
568 nk IF (x=9 AND drive$="d0: ") OR (k=10 AND drive$="d1 " THE
N
569 eK2 c$=file$(0)
570 w1 GOSUB LiesDiskName-WBenchToBack
571 Pv IF c$=file$(0) AND disk$=drive$ THEN RETURN
572 t9 disk$=drive$
573 qZ1 ELSE
574 KU2 MID$(drive$,3,1)=CHR$(46+ABS(VAL MID$(drive$,3,1)-1))
575 jP c$=disk$ disk$ drive$ x=0 GOSUB LiesDiskName-WBenchToBac
k
576 5n IF x=-1 THEN disk$=c$ drive$="d0:":RETURN
577 FB0 END IF
578 E1 COLOR 6,0:Gotoxy 4,6:POKE WINDOW(8)+56,2
579 rQ Printe file$(0)+SPACES(90) POKE WINDOW(8)+56,0
580 n1 GOSUB NeuesDir
581 jL RETURN
582 sh SelFile
583 J5 y=(my 113)\8 x=y
584 MR IF y>fanz%-1 THEN RETURN
585 Y6 FileLoop:
586 O2 IF y>fanz%-1 THEN y=fanz%-1
587 Cb IF x<>y THEN
588 Ha2 COLOR 6,0:LOCATE 15+x,2
589 OM Printe file$(efile+x)+SPACES(22-LEN(file$(efile+x))):x=y
590 SL0 END IF
591 j6 COLOR 0,6
592 GE LOCATE 15+y,2 Printe file$(efile+y)+SPACES(22-LEN(file$(ef
ile+y)))
593 xV mx=MOUSE(0):my=MOUSE(1):my=MOUSE(2)
594 oL IF mx>x1 AND mx<x2 AND my>y1 AND my<y2 THEN y=(my 113
\8
595 5X IF FN Maus=0 THEN FileLoop
596 P1 COLOR 6,0:LOCATE 15+x,2
597 gd Printe file$(efile+x)+SPACES(22-LEN(file$(efile+x)))
598 91 namen$(num)=file$(efile+x)
599 oF Gotoxy 4,202:Printe namen$(num)+SPACES(22-LEN(namen$(num)))
600 2e RETURN
601 SE SelDir
602 A7 IF RIGHT$(disk$,1," ") THEN i=1 j=5 ELSE i=0 j=6
603 EE y=(my-41)\8+1:x=y
604 gk IF y>danz THEN RETURN
605 21 DirLoop:
606 AW IF edir+y>danz THEN y=danz-edir
607 Wv IF x<>y THEN
608 bH2 COLOR 6,0:LOCATE j+x,2
609 qY Printe dir$(edir+x)+SPACES(22-LEN(dir$(edir+x))):x=y
610 mF0 END IF
611 jQ COLOR 0,6
612 nb LOCATE j+y,2 Printe dir$(edir+y)+SPACES(22-LEN(dir$(edir+y
)))
613 Hp mx=MOUSE(0):mx=MOUSE(1):my=MOUSE(2)
614 Kx IF mx>x1 AND mx<x2 AND my>y1 AND my<y2 THEN y=(my-41)\
8+
615 OK IF FN Maus=0 THEN DirLoop
616 jP COLOR 6,0:LOCATE j+x,2
617 7B Printe dir$(edir+x)+SPACES(22-LEN(dir$(edir+x)))
618 Nh IF edir+x=0 THEN
619 lA2 WHILE RIGHT$(disk$,1)<>"/" AND RIGHT$(disk$,1)<>": "
620 lJ4 disk$ LEFT$(disk$,LEN(disk$)-1)
621 SG2 WEND
622 ca IF LEN(disk$)<>4 THEN disk$=LEFT$(disk$,LEN(disk$)-1)
623 eH1 ELSE
624 tN2 IF RIGHT$(disk$,1)<>": " THEN disk$ disk$+"/"
625 dF disk$=disk$+dir$ edir+x,
626 2v0 END IF
627 ua c$=file$(0) GOSUB LiesDiskName-WBenchToBack
628 pM IF c$<>file$(0) THEN
629 qm2 Meldung "Falsche Disk in Laufwerk "+MID$(drive$,3,1),4
630 VV file$(0)=c$,GOSUB Abbruch k 8
631 K9 RETURN
632 BlQ END IF
633 Cg NeuesDir:
634 oN Gotoxy 4,30:Printe disk$+SPACES(40)
635 h6 FOR i=1 TO 8
636 552 LOCATE 14+i,2:Printe SPACES(22)-LOCATE 5+i,2:Printe SPAC
ES(22)
637 MR0 NEXT
638 fu FOR x=1 TO 3:namen$(x)="",NEXT
639 St Gotoxy 4,202:Printe namen$(num)+SPACES(22-LEN(namen$(num)))
640 kE GOSUB LeseDir-WBenchToBack
641 x5 edir=0 efile=1
642 7A GOSUB DirAnzeige.GOSUB FileAnzeige
643 jL RETURN
644 YP SaveFile
645 NK IF num<>2 THEN RETURN
646 yn Gotoxy 4,202:Eingabe 22,6,namen$(num)
647 nP RETURN
648 Ft DirDown
649 71 Invertieren
650 4e DD
651 Z22 IF danz-edir>8 THEN edir=edir+1:GOSUB DirAnzeige
652 2K0 IF FN Maus=0 THEN DD
653 B5 Invertieren
654 aW RETURN
655 Pb DirUp:
656 EB Invertieren
657 Or DU:
658 x82 If edir>0 THEN edir=edir-1:GOSUB DirAnzeige
659 n20 IF FN Maus=0 THEN DU
660 IC Invertieren
661 id RETURN
662 HR FileDown
663 LF Invertieren
664 Wy FD.
665 lU2 IF fanz%-efile>7 THEN efile=efile+1:GOSUB FileAnzeige
666 2K0 IF FN Maus=0 THEN FD
667 PJ Invertieren
668 Sk RETURN
669 xP FileUp
670 SM Invertieren
671 IS FU
672 TT2 IF efile>1 THEN efile=efile-1:GOSUB FileAnzeige
673 BPO IF FN Maus=0 THEN FU
674 WQ Invertieren
675 Fr RETURN
676 lJ DirAnzeige
677 W3 IF RIGHT$(disk$,1)=" " THEN k1=1 ELSE k1=0
678 No IF danz<8 THEN h=danz ELSE h=8
679 r2 LOCATE 5,2
680 yL COLOR 6,0
681 Oh FOR i=k1+edir TO edir+h
682 292 LOCATE CSRLIN+1,2:Printe dir$(i)+SPACES(22-LEN(dir$(i)))
683 6B0 NEXT
684 OU RETURN
685 2a FileAnzeige
686 sC IF fanz%<8 THEN h=fanz% ELSE h=8
687 NI LOCATE 14,2
688 6T COLOR 6,0
689 BM FOR i=efile TO h+efile
690 rT2 LOCATE CSRLIN+1,2:Printe file$(i)+SPACES(22-LEN(file$(i)
))
691 EJO NEXT
692 W6 RETURN
693 OQ Ok
694 bh Neusetzen x1,y1,x2,y2,farbe%,wort$,c$ file$,0,

```

Listing 2. Ein Spiel mit »allem Drum und Dran«:
»Buh-Au-Construction-Set«. Bitte mit dem Checksummer
 (Seite 143) eingeben (Fortsetzung).

```

695 x2 GOSUB LiesDiskName WBenchToBack
696 v5 IF c$ <> file$(0) THEN
697 w52 Meldung "Falsche Disk in Laufwerk "+MID$(drive$,3,1)+",4
698 Kq file$(0)=c$:GOSUB Abbruch
699 dF RETURN
700 E70 END IF
701 m5 IF namen$(num)=" THEN
702 DY2 Meldung "Kein Filename",4:GOSUB Abbruch
703 uJ RETURN
704 l80 END IF
705 wL ON num GOSUB Laden,Speichern,DelFile
706 na flag%=3
707 lN RETURN
708 Wm Speichern
709 lR x=0
710 UD FOR i=1 TO fanz%
711 oL2 IF UCASE$(namen$(num))=UCASE$(file$(i)) THEN x=-1:1-fanz%
712 z00 NEXT
713 q0 IF x THEN GOSUB Sicher2:IF NOT ja THEN RETURN
714 hq OPEN disk$+namen$(num)+".fld" FOR OUTPUT AS #1
715 R02 FOR y=0 TO 14
716 hn4 FOR x=0 TO 14:WRITE#1,feld$(x,y),init$(x,y):NEXT
717 aj2 NEXT
718 Bc WRITE#1,zeiger,anzahl,gesamt
719 v10 CLOSE#1
720 fr KILL disk$+namen$(num)+".fld.info"
721 wL WBenchToBack:WINDOW CLOSE 3:h=0
722 gP FOR i=1 TO fanz%
723 l02 IF UCASE$(namen$(num))=UCASE$(file$(i)) THEN h=-1
724 q0 NEXT
725 lb IF h THEN RETURN
726 kl fanz%=fanz%+1 file$,fanz%=namen$(num)
727 OD GOSUB FileSort
728 G1 RETURN
729 f1 Laden.
730 lV OPEN disk$+namen$(num)+".fld" FOR INPUT AS #1
731 4L2 WINDOW CLOSE 3
732 lR FOR y=0 TO 14
733 dG4 FOR x=0 TO 14
734 a16 INPUT #1,feld$(x,y),init$(x,y):GOSUB Kaestchen
735 w14 NEXT
736 x22 NEXT
737 p8 INPUT#1,zeiger,anzahl,gesamt
738 ob0 CLOSE#1
739 lt RETURN
740 wW DelFile
741 op KILL disk$+namen$(num)+".fld"
742 OJ FOR i=1 TO fanz%
743 BR2 IF file$(i)=namen$(num) THEN h=i
744 5A0 NEXT
745 u5 FOR i=h TO fanz%:SWAP file$(i+1),file$(i):NEXT
746 hq IF namen$(num)=namen$(1) THEN namen$(1)="
747 oE namen$(num)="" :fanz%=fanz%-1:WINDOW CLOSE 3
748 Q2 RETURN
749 lV Abbruch.
750 M4 WINDOW CLOSE 3
751 SA IF flag%=14 THEN flag%=22 ELSE flag%=3
752 U6 RETURN
753 qF Quit.
754 ag Neusetzen x1,y1,x2,y2,farbe$,wort$
755 J3 GOSUB Sicher
756 aw IF NOT ja THEN flag%=1:RETURN
757 th SCREEN CLOSE 1:WINDOW CLOSE 2:FreeMem info$,252k
758 jt END
759 j4 Spielerauswahl:
760 gm Neusetzen x1,y1,x2,y2,farbe$,wort$
761 bP WINDOW 3,,(200,64)-(440,160),16,1
762 bz RESTORE Daten5
763 a2 IF spanz% <> y2\50 THEN
764 b42 FOR i=1 TO 3:spielername$(i)="" :NEXT
765 hA0 END IF
766 JT spanz%=y2\50:Zeichnen spanz%+1:COLOR 6,0
767 KL FOR y=1 TO spanz%
768 yq2 LOCATE 2*y+INT(y/1.5),2
769 5A IF spanz%=1 THEN PRINT "Name "; ELSE PRINT "Name",y," ",
770 Mm Printe spielername$(y)
771 wB0 NEXT
772 JF k=0
773 s4 FOR i=1 TO spanz%
774 9t2 y=2*i+INT(1/1.5):LOCATE y,10:Zingabe 19,6,spielername$(i)

```

```

775 9h IF MOUSE(0) THEN i=spanz%:k=-1
776 qG0 NEXT
777 7D IF k THEN GOSUB MouseTest
778 7E RETURN
779 7S SpielerEingabe.
780 5J j=y2\18
781 JY IF j>spanz% THEN RETURN
782 49 j=2*i+INT(j/1.5) LOCATE y,10:Zingabe 19,6,spielername$(j)
783 M1 IF MOUSE(0) THEN GOSUB MouseTest
784 Oa RETURN
785 h5 Kaestchen
786 7B x3=300-x*18 x4=x3+16 y3=y*12 y4=y3+10
787 7n IF feld$(x,y)=-1 THEN
788 72 LINE(x3,y3)-(x4,y4),6,bf
789 31 ELSE
790 72 LINE(x3,y3)-(x4,y4),2,bf
791 hA0 END IF
792 7K RETURN
793 7M Endauswertung.
794 g 7M SE OFF CLS
795 eo IF komp THEN komp=0:spanz%=1
796 x0 Meldung spielername$(zahl)+" hat gewonnen !!!",zahl+2
797 7D COLOR 6,0:CLS:LOCATE 5,2:Printe "Die Punkte:"
798 7T FOR i=1 TO spanz%
799 722 j=(min$ 1 *60+sec$ 1)
800 7C IF wert$(i)>0 THEN punkte$(i)=INT((wert$(i)/gesamt*anza
hl*(30-(j/100)))*1.536421) ELSE punkte$(i)=j+5
801 Op IF punkte$(i)<=0 THEN punkte$(i)=5
802 7J COLOR i+2,0:LOCATE j+2*1,4
803 7a Printe spielername$(i)+SPACE$(20-LEN(spielername$(i)))+
:"
804 eL PRINT USING "#####":punkte$(i)
805 490 NEXT
806 7P 7D CLS h=0
807 7G LOCATE 17,27:Printe "Linke Maustaste dr"+CHR$(252)+"oken !
808 7x CALL WaitForClick:CLS:x=0
809 7e FOR i=1 TO spanz%
810 71 IF punkte$(i)>hsc$(13) THEN
811 7Y2 hsc$(13)=punkte$(i) hsc$(13)=spielername$(i):j=13:x=-1
812 77 WHILE hsc$(j)>hsc$(j-1) AND j>1
813 7a4 SWAP hsc$(j),hsc$(j-1):SWAP hsc$(j),hsc$(j-1):j=j-1
814 7N2 WEND
815 7TO ELSE
816 722 IF punkte$(i)<lsc$(13) THEN
817 724 lsc$(13)=punkte$(i) lsc$(13)=spielername$(i):j=13:x=-1
818 7q WHILE lsc$(j)<lsc$(j-1)
819 7a6 SWAP lsc$(j),lsc$(j-1):SWAP lsc$(j),lsc$(j-1):j=j-1
82 7T4 WEND
821 722 END IF
822 70 END IF
823 7N NEXT
824 7T IF x=-1 THEN GOSUB SoSpeichern
825 7y Farbenloeschen
826 7b COLOR 3,0:LOCATE 2,8:Printe "13 der Besten..."
827 73 LOCATE 2,47:Printe "und Schlechtesten "
828 7X LINE (0,16)-(640,18),bf.LINE (320,0)-(321,198),bf
829 71 FOR i=1 TO 13
830 722 j=25+1*13:COLOR 2,0:Gotoxy 8,j:PRINT USING "#####":j1
831 71 PRINT ".":COLOR 5,0:Gotoxy 64,j:Printe hsc$(i)
832 73 Gotoxy 232,j:PRINT USING "#####":hsc$(i);
833 73 COLOR 6,0:Gotoxy 376,j:Printe lsc$(i)
834 79 Gotoxy 544,j:PRINT USING "#####":lsc$(i)
835 7D j=29+1*13 LINE(0,j,- 640,j),7
836 7e0 NEXT
837 7k COLOR 4,0:LOCATE 28,27
838 7R Printe "Linke Maustaste dr"+CHR$(252)+"oken !! "
839 7s call Farbensetzen :call WaitForClick :CLS
840 7a GOSUB NeustartInitialisierungsRoutine:GOSUB Aufbau1
841 7X RETURN
842 7Q NeustartInitialisierungsRoutine
843 7E ERASE wert$,feldfarbe$:DIM wert$(3),feldfarbe$(14,14)
844 75 FOR y=0 TO 14
845 742 FOR x=0 TO 14
846 7D4 IF feld$(x,y)>0 THEN feld$(x,y)=0
847 7p2 NEXT
848 7q0 NEXT
849 7e null=0:flag%=1
850 7F FOR i=1 TO 3
851 7A2 IF huc$(i)=-1 THEN spielername$(i)=""
852 7e0 NEXT
853 7J RETURN

```



```

854 wo LeseDir:
855 SD FOR i=0 TO fanz%:file$(i)="" NEXT
856 eo FOR i=0 TO danz:dir$(i)="" NEXT
857 Tj GOSUB LiesDiskName
858 Do fanz%=0 danz=0 dir$(0)="/" Parent"ex&=lock&(SADD(a$),-2)
859 nb dummy&=Examine&(ex&,info&) dummy&=ExNext&(ex&,info&)
860 Q WHILE dummy&<>0
861 v2 fanz%=fanz%+1 file$(fanz%)="" i=0
862 tY GOSUB LiesName
863 Z IF j=2 THEN danz=danz+1:dir$(danz)=file$(fanz%)
864 QU IF RIGHT$(file$(fanz%),4)<>" .fld" THEN
865 p4 file$(fanz%)="" fanz%=fanz%-1
866 ZIF ELSE
867 A14 file$(fanz%)=LEFT$(file$(fanz%),LEN(file$(fanz%))-4,
868 wp2 END IF
869 e dummy&=ExNext&(ex&,info&)
870 THO WEND
871 pR FOR i=1 TO danz
872 c2 FOR j=1+1 TO danz
873 K4 IF UCASE$(dir$(j))<UCASE$(dir$(i)) THEN SWAP dir$(j),
dir$(i)
874 BG2 NEXT
875 CHO NEXT
876 KX FileSort
877 Su FOR i=1 TO fanz%
878 WJ2 FOR j=1+1 TO fanz%
879 tn4 IF UCASE$(file$(j))<UCASE$(file$(i)) THEN SWAP file$(
j),file$(i)
880 HM2 NEXT
881 INO NEXT
882 ye ex&=UnLock&(ex&)
883 bD RETURN
884 Yc LiesDiskName:
885 D9 a$=drive$+CHR$(0):ex&=lock&(SADD(a$),-2)
886 Qr IF ex&=0 THEN x=-1
887 6d dummy&=Examine&(ex&,info&) i=0 file$(0)="" h=fanz% fanz%=0
888 8a GOSUB LiesName:fanx%=h-a$=disk$+CHR$(0) ex&=UnLock&(ex&)
889 hJ RETURN
890 HT LiesName
891 yf j=PEEK info&+7) b$=CHR$(PEEK(info&+i+8))
892 6q IF ASC(b$)=0 THEN RETURN
893 0e file$(fanz%)=file$ fanz%+b$ i=i+1
894 vg GOTO LiesName
895 fJ SoLaden:
896 aw OPEN UrDisk$+".SCORES.bas" FOR INPUT AS 1
897 K3 FOR i=1 TO 13:INPUT #1,hsc$(i),hsc$(i),lsc$(i),lsc$(i) NEX
T
898 I2 CLOSE 1
899 rT RETURN
900 Lp SoSpeichern:
901 i7 OPEN UrDisk$+".SCORES.bas" FOR OUTPUT AS 1
902 do FOR i=1 TO 13:WRITE #1,hsc$(i),hsc$(i),lsc$(i),lsc$(i) NE
XT
903 2R CLOSE 1-KILL UrDisk$+".SCORES.bas.info"
904 z0 WrenchToBack
905 xZ RETURN
906 TA HighScores:
907 M9 IF NOT ScoreFehler% THEN GOSUB SoLaden:RETURN
908 FV RESTORE HighDaten
909 2q FOR i=1 TO 13:READ hsc$(i),hsc$(i),lsc$(i),lsc$(i) NEXT
910 hn GOSUB SoSpeichern
911 jf RETURN
912 pj HighDaten
913 81 DATA K. Schreck,4000,Murphy,10,M. Boney,3000,Murphy,20,AMIG
A,1000
914 iJ DATA Murphy,30,AMIGA,800,Murphy,40,AMIGA,700,Murphy,50,AMI
GA,650
915 8K DATA Murphy,70,AMIGA,600,Murphy,90,AMIGA,550,Murphy,110,AM
IGA,500
916 Nv DATA Murphy,130,AMIGA,450,Murphy,150,AMIGA,400,Murphy,170,
AMIGA,300
917 tJ DATA Murphy,190,AMIGA,200,Murphy,200
918 m8 SUB Eingabe(max%,farbe%,eing%) STATIC
919 Wo COLOR farbe%,0:Print e ing$ COLOR 0,farbe%
920 6c PRINT " ";:COLOR farbe%,0:PRINT " ";CHR$(8);
921 Gp WHILE INKEY$<>"":WEND
922 4 WHILE MOUSE(0):WEND
923 S3 WHILE a$<>CHR$(13) AND MOUSE(0)=0
924 LR2 a$="" a$=INKEY$
925 +X IF a$<>"": THEN
926 Qt4 IF ASC(a$)>31 AND LEN(eing$)<max% THEN
927 ex6 eing$=eing$+a$:PRINT CHR$(8),a$,

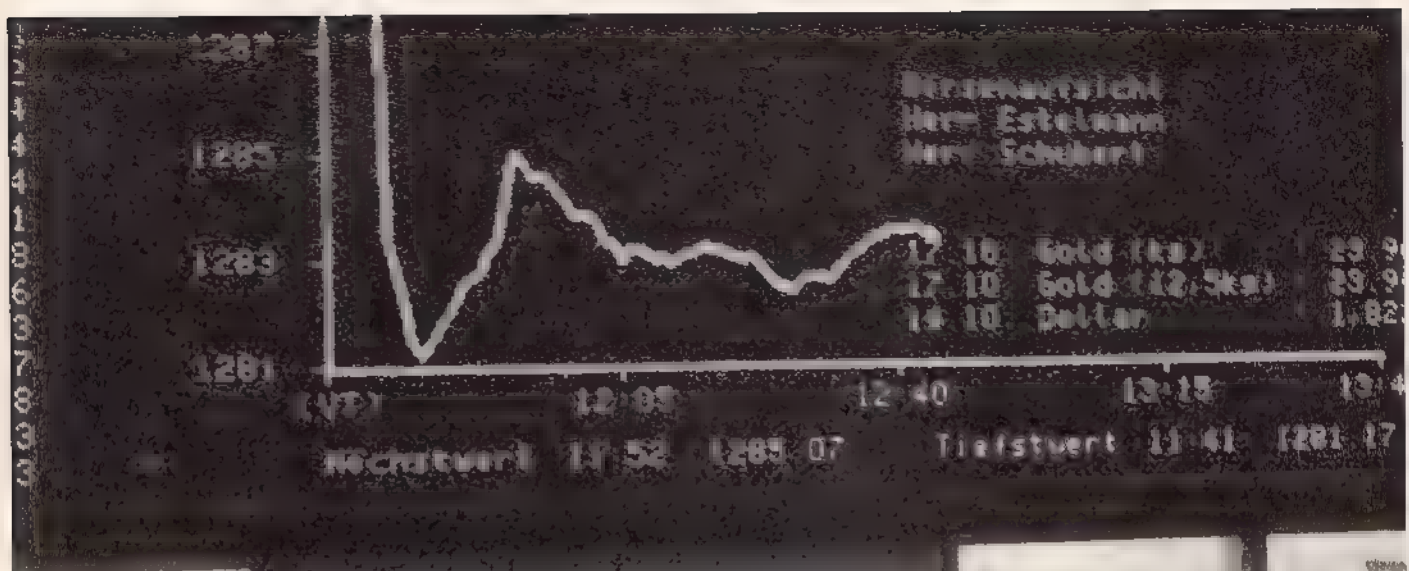
```

```

928 2N COLOR 0,farbe%:PRINT " ";:COLOR farbe%,0
929 vo4 END IF
930 Io IF a$=CHR$(8) AND LEN(eing$)>0 THEN
931 7q6 PRINT CHR$(8),CHR$(8);:COLOR 0,farbe%
932 Qd PRINT " ";:COLOR farbe%,0:PRINT " ";CHR$(8);
933 6U eing$=LEFT$(eing$,LEN(eing$)-1)
934 Ot4 END IF
935 1u2 END IF
936 XLQ WEND
937 Lp PRINT CHR$(8),
938 AC END SUB
939 X2 SUB Zeichnen(max%) STATIC
940 xM FOR i=1 TO max%
941 XX2 READ x1,y1,x2,y2,flag%,farbe%,wort$
942 zL LINE (x1,y1)-(x2,y2),farbe%,b:COLOR farbe%,0
943 VC x=x1+x2 x1=LEN(wort$)*8\2:y=y1+(y2-y1+4)\2
944 Je Gotoxy x,y:Printe wort$
945 KPO NEXT
946 IK END SUB
947 0e SUB Zeichnen2(x1,y1,x2,y2,farbe%,wort$) STATIC
948 mE LINE (x1,y1)-(x2,y2),farbe%,b:COLOR 0,farbe%
949 bJ x=x1+(x2-x1-LEN(wort$)*8)\2:y=y1+(y2-y1+4)\2
950 4F Gotoxy x,y:Printe wort$ COLOR farbe%,0
951 S7 WhileMousePressed
952 OQ END SUB
953 Ov SUB Printe (a$) STATIC
954 pB Text WINDOW(8),SADD(a$),LEN(a$)
955 RT END SUB
956 Qg SUB Meldung(Wertung$,exfarbe%) STATIC
957 oJ x=((640-((LEN(Wertung$)+2)*8))\2)
958 mK WINDOW 3,,(x,80)-(x+LEN(Wertung$)*8)+16,96,16,1
959 4I WINDOW OUTPUT 3
960 eQ PAINT (0,0),exfarbe%:COLOR 0,exfarbe%:PRINT
961 az Printe " "+Wertung$+" "
962 GY IF exfarbe%=2 THEN
963 TT2 j=.TIMER
964 F9 WHILE TIMER<j+1:WEND
965 At1 ELSE
966 1K2 call WaitForClick
967 XQO END IF
968 ta WINDOW CLOSE 3
969 fh END SUB
970 Hs SUB Neusetzen(x1,y1,x2,y2,farbe%,wort$) STATIC
971 mE LINE (x1+1,y1+1)-(x2-1,y2-1),0,bf:COLOR farbe%,0
972 yf x=x1+ x2-x1-LEN(wort$)*8\2:y=y1+(y2-y1+4)\2
973 m7 Gotoxy x,y:Printe wort$
974 km END SUB
975 Y1 SUB FileKasten(y,farbe%) STATIC
976 am LINE (4,38+y)-(186,105+y),farbe%,b
977 61 LINE (186,38+y)-(210,105+y),farbe%,b
978 p9 LINE (186,52+y)-(210,91+y),farbe%,b
979 yE AREA (198,42+y):AREA (192,48+y):AREA (204,48+y).AREAFILL
980 rw AREA (198,101+y):AREA (192,95+y):AREA (204,95+y):AREAFILL
981 rt END SUB
982 JG SUB Gotoxy(x,y) STATIC
983 BL Move WINDOW(8),x,y
984 LW END SUB
985 OY SUB WaitForClick STATIC
986 Hn WHILE FN Maus=0:WEND
987 xz END SUB
988 4s SUB WhileMousePressed STATIC
989 8c WHILE FN Maus=0:WEND
990 O2 END SUB
991 rr SUB Invertieren STATIC
992 rr SHARED x1,y1,x2,y2
993 wE SetDrMd WINDOW(8),2:LINE (x1,y1)-(x2,y2),6,bf
994 FE SetDrMd WINDOW(8),1
995 57 END SUB
996 41 SUB Farbansetzen STATIC
997 1k PALETTE 0,0,0,0:PALETTE 1,0,0,0:PALETTE 2,.4,.6,1
998 NF PALETTE 3,.33,.87,0:PALETTE 4,.93,.2,0:PALETTE 5,1,1,.13
999 1b PALETTE 6, .73,.73, .73:PALETTE 7,.8,.321,.95
1000 AC END SUB
1001 wI SUB Farbenloeschen STATIC
1002 T8 FOR x=0 TO 7:PALETTE x,0,0,0:NEXT
1003 DF END SUB
(C) 1989 M&T

```

Listing 2. Ein Spiel mit »allem Drum und Dran«:
»Buh-Au-Construction-Set«. Bitte mit dem Checksummer
(Seite 143) eingeben (Schluß).



BÖRSENFIEBER AM AMIGA

Die Börse — dieser Begriff setzt Assoziationen frei
Der eine denkt an Spekulationsgewinn
und Husarenstreiche, der andere an Crash
und Verluste. Erleben Sie die Faszination
der Börse »live« auf Ihrem Amiga
— und das ohne finanzielles Risiko.

Kaufen und Verkaufen, durch geschicktes Manipulieren den Gegner übernahmreif machen. Dieses Thema stammt nicht aus dem Wallstreet-Journal, sondern von einer turbulenten Börsensimulation, die so manchen eingefleischten Broker ins Schwitzen bringen kann. »Broker« (Listing 1) ist ein Spiel für zwei bis vier Teilnehmer. Durch geschickte Aktionen versucht jeder, den Kurs der eigenen Aktien möglichst hoch zu treiben, und den der Mitspieler so tief wie möglich zu drücken.

Wer zuerst eine vorher vereinbarte Summe erreicht, ist Sieger und kann sich belohnen als Börsen-Profi fühlen. Sie benötigen allerdings sehr viel planerisches Geschick und ein wenig Glück, um gegen die Konkurrenz zu bestehen.

Alle wichtigen Kommandos werden mit der rechten Maustaste über Pull-Down-Menüs oder wahlweise über Tastenkombinationen (Shortcuts) aufgerufen, die man sich mit dem Menüpunkt **Help** (Bild 1) ansehen kann. Um ein Spiel zu beginnen, wählen Sie bitte **Neues Spiel** an. Nach Eingabe aller Spielernamen werden Sie aufgefordert, das Startkapital und den Betrag festzusetzen, bei dessen Erreichen das Spiel beendet wird. Der Mindestbetrag, der vom Sieger zu erreichen ist, muß mindestens zehnmal so groß sein wie das Startkapital.

Broker's Basar

In jeder neuen Runde haben Sie fünf Handlungspunkte zur Verfügung, die Sie gegen Aktien an der Börse tauschen können. Je nach Art der Aktion werden unterschiedliche Punktzahlen abgezogen. So kostet jeder Aktienkauf und -verkauf jeweils zwei Punkte, Manipulationen der Kurse (siehe unten) zwischen einem und drei Punkten. Daneben können Sie verschiedene Dienstleistungen in Anspruch nehmen, wobei die meisten bares Geld kosten. So können Sie sich für 60 Mark die Kursentwicklung ausgewählter Aktien ansehen oder für 30 Mark einen Vergleich des Kapitals aller Spieler ausgeben lassen. 40 Mark müssen Sie investieren, wenn Sie Informationen über die Aktien Ihrer Gegner wollen.

Sie sollten dabei bedenken: Jede Information schmälert zwar Ihr Bankkonto, aber eine

BROKER - Das Börsenspiel von Sebastian Peatz

Menüpunkt	Taste	Erklärung	DWHP
Laden	ALT l	Ein Spiel wird von Disk geladen	
Speichern	ALT s	Ein Spiel wird abgespeichert	
Neues Spiel	ALT n	Es wird ein neues Spiel begonnen	
Quit	ALT q	Spielende ohne die Spielstände abzuspeichern	
Börsenaktion	a	Der Spieler kann aktiv die Börsenkurse durch geschicktes Handeln beeinflussen	20 1-3
Börsenkurse	b	Es werden die aktuellen Börsenkurse angezeigt	- -
Charts	c	Aktienkurse können graphisch dargestellt werden	60 -
Aktienkauf	k	Aktien können zum Tageskurs gekauft werden	5% 2
Aktienverk.	v	Aktien können zum Tageskurs verkauft werden	5% 2
Kapitalliste	l	Eigenes Kapital (Aktienkapital und bar)	- -
Kapitalvergl.	r	Kapital aller Spieler	30 -
Aktienliste	i	Auflistung der eigenen Aktien (Kurse, Stück)	- -
Aktienvergl.	g	Aktien aller Spieler (Stück)	40 -
Nächster	n	Der nächste Spieler ist an der Reihe	
Übersicht	u	Alle wichtigen Daten zum Spiel im Überblick	
Info	Help	Diese Seite	

Bild 1. Diese Übersicht erreichen Sie mit der »Help«-Taste oder mit Help aus dem Info-Menü

Fehlinvestition kann weit wertvollere Handlungspunkte kosten. Sie können mit dem Menüpunkt »Börsenaktion« Aktienkurse teilweise gezielt beeinflussen. Voraussetzung hierfür ist, daß Sie die Besitzverhältnisse aller Spieler genau kennen: Wie viele Brauereiaktien hatte doch gleich die Konkurrenz, waren Versicherungsaktien nicht eben billiger geworden?

Wenn Sie keine Handlungspunkte mehr haben, können Sie nur noch Informationen einholen oder an den nächsten Spieler abgeben (mit dem Menüpunkt »Nächster«). Bei jedem Wechsel werden die Schulden, die sich im Verlauf der aktuellen Runde angesammelt haben, mit dem Barkapital verrechnet.

Das Spiel ist zu Ende, wenn das Nettokapital eines Spielers (Bargeld plus in Aktien angelegtes Kapital abzüglich Schulden) unter 100 Mark fällt oder ein Broker den festgelegten Spielendbetrag erreicht. Sind dabei noch Schulden vorhanden, so werden erst diese verrechnet. Sind die Schulden eines Spielers größer als die Hälfte des Startkapitals, wird das Spiel ebenfalls beendet.

Spekulieren Sie mit

Kapital erwirtschaftet der Broker, indem er Aktien zu einem möglichst billigen Kurs erwirbt (Bild 2) und diese zu einem hohen Preis wieder verkauft. Bei Kauf und Verkauf werden jeweils 5 Prozent des

Aktienkauf

Name: Boesky Barkapital: 3704 DM

Welche Aktie möchten Sie kaufen? <0> Keine

Nr.	AG	Vorrätig	Kurs
< 1 >	Bauwerte	1732	167
< 2 >	Textil	1452	149
< 3 >	Brauerei	1408	278
< 4 >	Kosmetik	1250	282
< 5 >	Elektrizität	1250	428
< 6 >	Chemie	1000	428
< 7 >	Pharmazie	1000	604
< 8 >	Maschinenbau	750	574
< 9 >	Kaufhaus	750	720
< 10 >	Computer	500	760
< 11 >	Automobil	500	878
< 12 >	Banken	250	878
< 13 >	Versicherung	250	760

Bild 2. Das Schaufenster für den Aktieneinkauf

Umsatzes als Maklergebühren vom Bargeld abgezogen und zwei Handlungspunkte verbraucht. Damit der Spieler

nicht den Überblick verliert, zeigt der Computer beim Verkauf der Aktien den Ankaukurs und den entsprechenden Gewinn oder Verlust an.

Beim Menüpunkt Börsenaktion geht es richtig los. Hier spielen sich alle Kursveränderungen ab. Allerdings haben Sie nicht unbegrenzt Einfluß auf die Kurse. Beim Aufruf einer Börsenaktion bestimmt der Zufall, welche Manipulationen ermöglicht werden. Die ganze Handlungspalette zeigt sich bei »Manual Activity II« (Bild 3). Hier können Sie wählen, welche Manipulation ausgeführt werden soll. Dazu bekommen Sie noch eine kurze Erklärung zur Wirkung der einzelnen Punkte.

Broker

Broker sind Leute, deren Job es ist, an der Börse mit Aktien und Wertpapieren zu handeln. Dabei werden Kursdifferenzen ausgenutzt, um Gewinne zu erzielen. Mit verschiedenen Techniken ist es möglich, Kurse zu beeinflussen. So steigt der Kurs von Aktien häufig auf das Gerücht hin, die Firma würde von einer anderen übernommen. Solche Tendenzen geschickt auszunutzen, ist Aufgabe eines guten Brokers.



Bild 3. »Manual Activity II« zeigt Ihnen die ganze Trickkiste der Manipulation

So gibt es verschiedene Aktionen, die eine ausgewählte Aktie steigen oder fallen lassen und alle anderen umgekehrt beeinflussen. Wenn Sie Ihren Mitspielern das Leben schwer machen wollen, ist »Manual Activity I« für Sie das Richtige: Hier können Sie selbst eine Aktie bestimmen, die steigen und eine, die sinken soll. Außerdem sind die Veränderungen hier weit größer als bei den zufalls-gesteuerten Varianten.

Ebenfalls sehr große Veränderungen erzielen Sie mit »Multiple Action Flood«. Hier wird jede einzelne Aktie entweder stark auf- oder abgewertet. Fast schon harmlos ist dagegen der gelegentlich auftauchende Fiskus, das Finanzamt zeigt sich ungewohnt milde: Steuern werden — im Gegensatz zur Realität — nur auf das Barvermögen erhoben.

Notbremse für Fehler

Jede der zufallsgesteuerten Aktionen kostet Sie einen Handlungspunkt. Sie können Kursveränderungen allerdings auch noch aufhalten, beispielsweise, wenn Ihre Aktien durch eine zufällige Manipulation geschwächt oder die Ihrer Gegner gestärkt würden. In diesem Fall werden Ihnen statt einem gleich drei Handlungspunkte abgezogen. Sie sollten sich also genau überlegen, ob Sie eine Aktion wirklich rückgängig machen wollen. Für die wertvolleren Aktionen, bei denen Sie direkten Einfluß auf die ausgewählten Aktien haben oder die hohe Kursunterschiede mit sich bringen, werden Ih-

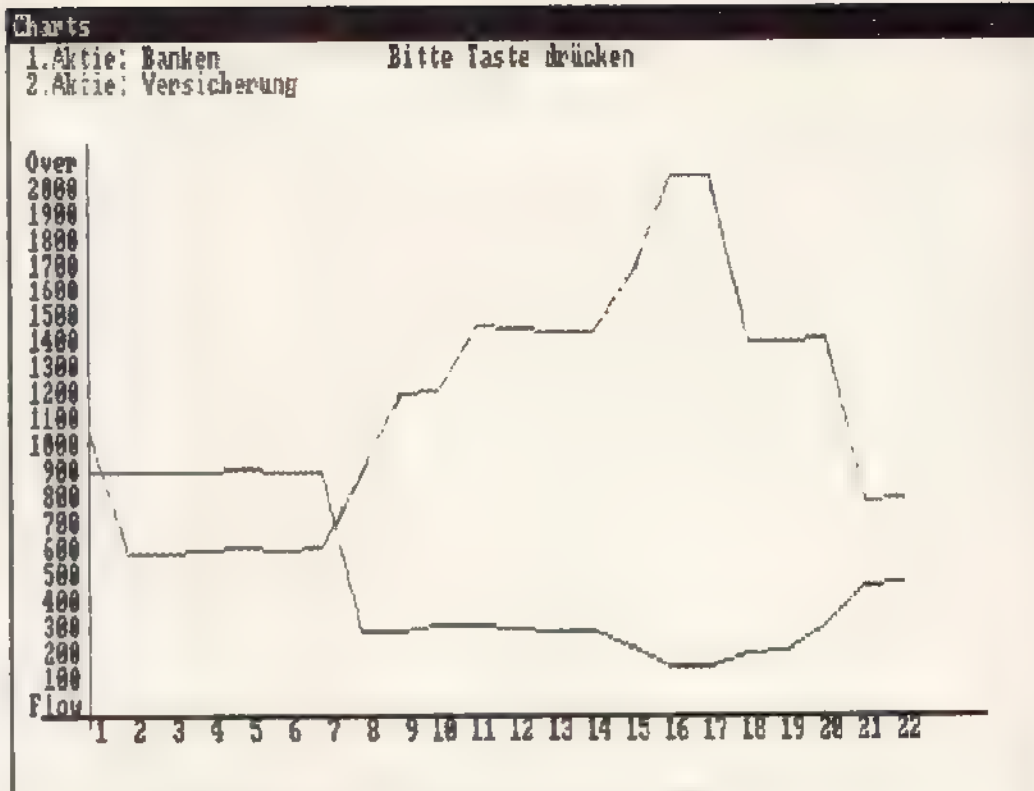


Bild 4. Ausgewählte Aktien können Sie als Grafik ausgeben lassen

Kapitalvergleich

Name	Bargeld	Aktienwert	Brutto	Schulden	Netto
Ivan	961	6364	7225	0	7225
Boesky	191	17303	17494	130	17364

Bild 5. Kapitalvergleich gibt Ihnen Informationen über Ihre Mitspieler

nen immer drei Punkte abgezogen.

Neben diesen Menüpunkten, die aktive Manipulationen zulassen, gibt es die Möglichkeit, Informationen einzuholen. Der Menüpunkt **Börsenkurse** öffnet ein Fenster, in dem die Kurse aller Aktiengesellschaften gezeigt werden. **Ekurs** bedeutet Einstandskurs und dient zur Bewertung des derzeitigen Aktienstands. Wer eine grafische Auswertung der Kursentwicklung sehen möchte, kommt bei **Charts** (Bild 4) auf seine Kosten. Es werden die letzten 22 Kursbewegungen von maximal zwei Aktien gleichzeitig grafisch dargestellt. Hier sollten Sie aber genau wissen, was Sie wollen, denn jede Grafik kostet 60 Mark.

Daneben können Sie sich über die jeweiligen Besitzverhältnisse informieren. **Kapitalliste** zeigt das aktuelle Kapital des Spielers, der am Zug ist, in Barwert und Aktienwert. **Ak-**

ziehungsweise 40 Mark (Aktienvergleich) für Gebühren ausgeben.

Mit dem Menüpunkt **Sichern** können Sie ein Spiel unterbrechen und den Spielstand auf Diskette sichern. Beim erneuten Start wählen Sie statt **Neues Spiel** einfach **Laden** und spielen dort weiter, wo Sie aufgehört haben.

Sie können hierfür zwei Verfahren anwenden: Wenn Sie für den gespeicherten Spielstand das Directory »Brokerdata« verwenden wollen, brauchen Sie im Programm nichts zu verändern. Sie geben einfach im CLI ein »makadir Brokerdata«. Wollen Sie dagegen ein eigenes Directory verwenden, dann müssen Sie den Pfadnamen bei den gekennzeichneten Stellen im Listing auf die richtige Diskette und Schublade umstellen (Zelle 76, 78, 87 sowie 97, 99 und 108).

Mit dem Menüpunkt **Quit** verlassen Sie Broker, ohne den Spielstand zu sichern.

Tippen Sie das Basic-Programm »Broker« ab und lassen Sie sich faszinieren von der Welt der schnellen Gewinne und Verluste, der Spekulation und der Übernahmen.

Erleben Sie das Auf und Ab der Börse, Hausse und Baisse, Erfolge und Crashes — und das alles, ohne Ihr Bankkonto zu belasten.

Christian Buchner/kn

Eingabehinweise

Tippen Sie Broker mit dem Checksummer ab. Beachten Sie dabei die Anleitung auf Seite 143. Starten Sie jetzt Amiga-Basic. Aktivieren Sie das Ausgabefenster und geben Sie
CLEAR ,40000,5000
ein. Laden Sie nun mit »Open« Ihr Programm und starten Sie es mit »Start«.

Aktienliste dagegen zeigt Ihren gesamten Aktienbesitz. Um sich Informationen über andere Spieler zu besorgen, muß man Gebühren bezahlen. **Kapitalvergleich** gibt das Kapital aller Spieler aus (Bild 5), **Aktienvergleich** den Besitzstand in Aktien.

Informationen über Ihren eigenen Besitz erhalten Sie kostenlos. Wünschen Sie dagegen Informationen über die anderen Spieler, so müssen Sie 30 Mark (Kapitalvergleich) be-

Programmname: Broker

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: Amiga-Basic

```

1 x00 ' *****
2 9S ' * BROKER - Das Börsenspiel von Sebastian Peetz *
3 up ' * (C) 1988 Markt & Technik *
4 OR ' *****

5 GP RANDOMIZE TIMER
6 17 OPTION BASE 1
7 2D Variablen und Screenaufbau
8 6r DIM sp$(4),Ak$(13),k$(4),kap(4),ka(4,13),x(13),z(13),v(13)
9 tw DIM st$(13),note$(13),w(4,13),gr(2)
10 S1 p=0 me2=0.0 req=0.0 endes=0 lo=C
11 zn SCREEN 2,640,255,4,2
12 sh WINDOW 2,"BROKER",(0,0)-(630,240),0,2
13 10 WINDOW OUTPUT 2
14 XU ON ERROR GOTO Fehler
15 fH FOR i=0 TO 15
16 5m1 READ a,b,c
17 3M PALETTE 1,a/15,b/15,c/15
18 jz0 NEXT i
19 eK DATA ,,,15,15,15,15,15,,,15,,,15,15,,15,8,8,,,15
20 q1 DATA 15,15,15,15,15,,10,5,3,,15,,,15,15,,15,15,15
21 fd LOCATE 15,25:PRINT "Bitte warten Sie einen Moment"
22 Tv Aktiendaten lesen
23 on RESTORE papiere
24 OF FOR i=1 TO 13:READ Ak$(i):NEXT i
25 6S Menüleiste 1
26 FL ml.
27 st1 RESTORE menua
28 xw READ titel:FOR i=1 TO titel
29 YR READ unter:FOR j=0 TO unter
30 jf2 READ statu:IF statu>2 THEN statu=1
31 e3 IF j=0 AND statu=2 THEN statu=1
32 Fs READ a$
33 j6 MENU i,j,statu,a$
34 2T1 NEXT j:NEXT i
35 7D0 CLS
36 SP1 ON MENU GOSUB split
37 y1 GOTO sch1
38 NU0 Menüleiste 2
39 fo m2:
40 GB1 RESTORE menub
41 A9 READ titel:FOR i=1 TO titel
42 le READ unter:FOR j=0 TO unter
43 7s2 READ statu:IF statu>2 THEN statu=1
44 OG IF j=0 AND statu=2 THEN statu=1
45 S5 READ a$
46 wJ MENU i,j,statu,a$
47 BA1 NEXT j:NEXT i:me2=1
48 KQ0 CLS
49 oK 'Mausabfrage zu Menüleiste 2
50 gd1 ON MENU GOSUB split
51 zd IF p THEN GOTO Kurse
52 D0 GOTO scnl
53 tQ0 'Hauptprogramm
54 In c
55 RX1 CLS
56 Zn IF hp>0 THEN GOSUB Hpu ELSE IF p THEN GOSUB NoHpu
57 8G0 sch1:
58 Aa1 MENU ON
59 wM a$=INKEY$:IF a$="" GOTO scnl
60 4JD split
61 XR1 ma=MENU(1):MENU OFF
62 91 ON MENU(0) GOTO Broker,Boerse,Aktien,Kapakt,Info
63 oS0 Broker:
64 xp1 ON nu GOTO lad,sav,neue,Erde
65 Yz0 Boerse
66 b11 ON nu GOTO Boersenk,Kurse,Charts
67 X20 Aktien:
68 Ke1 ON nu GOTO Aktkauf,Aktverk
69 H10 Kapakt:
70 V21 ON nu GOTO Kaplist,Kapvar,Aktlist,Aktver
71 e10 Info
72 Bw1 ON nu GOTO wech,Uebers,Help
73 Bm0 'Menüpunkt Laden
74 H41 lad
75 g02 WINDOW 4,"Daten laden",(0,10)-(300,180),0,2:COLOR 13,7:CLS
LOCATE 2,1
76 bd FILES "BrokerData" "Directorypfad ge

```

```

gebenfalls ändern!
77 6L PRINT:LINE INPUT "Dateiname: ";dat1$
78 fb OPEN "BrokerData/"+dat1$ FOR INPUT AS #1 'Dirpfad gegebenfalls ändern
79 KA INPUT #1,s,kapital,spielende,p,ap
80 8k FOR i=1 TO s:INPUT #1,sp$(i):INPUT #1,kap(i):INPUT #1,kre(i):NEXT i
81 eF FOR i=1 TO 13
82 ol3 INPUT #1,x(i):INPUT #1,v(i):INPUT #1,z(i):INPUT #1,ts(i):NEXT i
83 m22 NEXT i
84 13 FOR j=1 TO s
85 e93 FOR i=1 TO 13:INPUT #1,w(j,i):INPUT #1,ka(j,i):NEXT i
86 r82 NEXT j
87 13 CLOSE #1
88 1c FOR i=1 TO 13:ve=VAL(MID$(st$(i),1,4))
89 73 IF (ve<1000 AND ve>99) THEN st$(i)=" "+st$(i)
90 IF ve<99 THEN st$(i)=" "+st$(i)
91 122 NEXT i:me2 THEN GOTO k
92 rB WINDOW CLOSE 4:WINDOW OUTPUT 2
93 aK LOCATE 15,25:PRINT "Bitte warten Sie einen Moment":GOTO m
94 z80 'Menüpunkt Speichern
95 OC1 sav:
96 632 WINDOW 4,"Daten abspeichern",(0,10)-(300,180),0,2:COLOR 13,7:CLS:LOCATE 2,1
97 2Z FILES "BrokerData" 'Dirpfad gegebenefalls ändern!
98 qU PRINT:LINE INPUT "Dateiname: ";data$
99 vH OPEN "BrokerData/"+data$ FOR OUTPUT AS #1 'siehe oben
100 4r PRINT #1,s,kapital,spielende,p,ap
101 Hr FOR i=1 TO s:PRINT #1,sp$(i):PRINT #1,kap(i):PRINT #1,kre(i):NEXT i
102 za FOR i=1 TO 13
103 Dr3 PRINT #1,x(i):PRINT #1,v(i):PRINT #1,z(i):PRINT #1,ts(i):PRINT #1,note$(i)
104 7M2 NEXT i
105 60 FOR j=1 TO s
106 NF3 FOR i=1 TO 13:PRINT #1,w(j,i):PRINT #1,ka(j,i):NEXT i
107 072 NEXT j
108 G1 CLOSE #1:KILL "BrokerData/"+data$+".info" 'siehe oben
109 VM WINDOW CLOSE 4:WINDOW OUTPUT 2:GOTO Quit
110 KAO 'Menüpunkt Neues Spiel:
111 DO1 neue
112 Ah2 WINDOW 4,"Neues Spiel",(0,10)-(280,140),0,2:COLOR 15,5:CLS
113 ip1 vie:
114 en2 LOCATE 2,1:INPUT "Wieviele Mitspieler?";s:IF (s<2 OR s>4) THEN vie
115 lb PRINT:FOR i=1 TO s
116 y81 sp=
117 JQ2 COLOR 1,5
118 A2 LOCATE (2*i+2),1
119 eY PRINT "Name des ";i". Spielers "
120 yf COLOR 2,5
121 EB LINE INPUT sp$(i):l=LEN(sp$(i)):IF (l<1 OR l>10) THEN sp=
122 7f1 NEXT i
123 7A kap1
124 F2 LOCATE 15,5:LOCATE (2*i+3),1:INPUT "Startkapital (500-1200)";kapital
125 57 IF (kapital<500 OR kapital>12000) THEN kap1
126 FY1 spielende:
127 4h2 LOCATE (2*i+4),1:INPUT "Spielende (>=Kapital*10)";spielende
128 Ja IF (spielende<kapital*10 OR spielende>99999999) THEN sp1=
129 2 LOCATE 2,5:CLS:LOCATE 2,1
130 1c FOR i=1 TO s
131 Js+ PRINT "Spieler ";i " ";sp$(i) PRINT
132 V52 NEXT i:COLOR 5,2
133 gh PRINT:LOCATE ,2:PRINT "Startkapital: ";kapital;" DM"
134 qa LOCATE ,2:PRINT "Spielende bei: ";spielende;" DM"
135 1W COLOR 15,5:PRINT:PRINT:INPUT "Alles in Ordnung?";a$:IF (a$="n" OR a$="N") GOTO Ne1
136 7t IF a$="n" OR a$="N" THEN GOTO Ne1
137 7t1 a1
138 682 CLS:LOCATE 15,22:PRINT "Bitte warten - AMIGA generiert Dat"
139 v1 FOR ip=1 TO 13:st$(ip)="":NEXT i
140 2b FOR ip=1 TO 13:x(ip)=INT(1/24.5)*150

```

Listing 1. »Broker« geben Sie mit dem Checksummer (Seite 143) ein

```

141 Lj3 v(ip)=(A5S((ip<2)+(ip<4)+(ip<6)+(ip<8)+(ip<10)+(ip
142 Nr FOR io=1 TO 23
143 en4 h$="0000"+STR$(x(ip))
144 SD st$(ip)=st$(ip)+MID$(h$,LEN(h$)-3,4)
145 b83 NEXT io
146 gZ note$(ip)="BBR":s(ip)=INT(ip/2+.5)*150
147 gE2 NEXT ip
148 TO FOR i=1 TO s:kap(i)=kapital:kre(i)' C
149 7E3 FOR ip=1 TO 13:w(i,ip)=0:kz(i,ip)=C NEXT ip
150 r72 NEXT i
151 1X p=1:hp=5:IF me2 THEN CLS:GOTO Kurse
152 UO GOTO m2
153 IG0 'Menüpunkt Quit
154 vJ1 Ende.
155 h52 MENU RESET
156 1Q WINDOW CLOSE 2
157 OU SCREEN CLOSE 2
158 ID END
159 wk0 'Menüpunkt Börsenaktion
160 MD Boersena
161 QJ1 WINDOW 4,"Börsenaktion",{0,10}-{330,235},0,2:CLS
162 mk IF hp<3 THEN GOTO Notg
163 aD kre(p)=kre(p)+20:r1=INT(RND*13)+1
164 GP ON r1 GOTO Div1,HaBa,HaBa,HaBa,HaBa,Flood,HaBa,HaBa,HaBa,H
aBa,Ma1,Ma2,Fiskus
165 Ob0 Dividende/Aufwertung
166 Ky1 Div1:
167 aJ2 COLOR 0,15:CLS:PRINT:PRINT" Dividende/Aufwertung"
168 NR r1=INT(RND*15):IF r1=0 OR r1=14 THEN GOSUB Wahl:GOTO eing
169 GW GOTO wei
170 mni eing:
171 W82 LOCATE 2,1:INPUT" AG-Nummer Dividende/Aufwertung":r1
172 7P IF r1<1 OR r1>13 THEN eing
173 yP COLOR 0,15:CLS:PRINT:PRINT" Dividende/Aufwertung"
174 La1 wei:
175 F82 IF x(r1)<v(r1)THEN GOTO wei3
176 vv PRINT:PRINT" Dividende 4%":PRINT
177 xe PRINT" -----"
178 km PRINT" AG Diff. Alt Neu"
179 gD PRINT:PRINT TAB(2)Ak$(r1)TAB(18)INT(x(r1)/25)TAB(26)x(r1)
TAB(33)x(r1)-INT(x(r1)/25)
180 Oh PRINT" -----"
181 oW PRINT:PRINT" Spieler Haben $DM Gewinn"
182 V8 FOR ip=1 TO s:LOCATE 13+ip,2
183 5V3 PRINT sp$(ip)TAB(15)w(ip,r1)TAB(25)INT(x(r1)/25)TAB(32)v
(ip,r1)*INT(x(r1)/25)
184 ZK2 NEXT ip:PRINT
185 7T LOCATE 19,1:INPUT" Akzeptieren Sie";jn$:IF jn$="n" OR j
n$="N" THEN hp=hp-3:GOTO wei2
186 GF FOR ip=1 TO s:kap(ip)=kap(ip)+w(ip,r1)*INT(x(r1)/25):NEXT
ip
187 BQ x(r1)=x(r1)-INT(x(r1)/25):hp=hp-1
188 rA1 wei2:
189 m82 GOTO kk
190 yI1 wei3:
191 942 PRINT:PRINT" Aufwertung (Kurs=EKurs)" PRINT
192 sm PRINT" -----"
193 Bt PRINT" AG Aufw. Alt Neu"
194 G3 PRINT:PRINT TAB(2)Ak$(r1)TAB(18)z(r1)-x(r1)TAB(26)x(r1)TA
B 33 z(r1)
195 vp PRINT" -----"
196 ad PRINT:PRINT" Spieler Haben $DM Verlust"
197 sh FOR ip=1 TO s:LOCATE 13+ip,2
198 wk3 PRINT sp$(ip)TAB(15)w(ip,r1)TAB(24)z(r1)-x(r1)TAB(32)v(i
p,r1)*(z(r1)-x(r1))
199 hF IF w(ip,r1)=0 THEN wei4
200 9D IF kap(ip)>(w(ip,r1)*(x(r1)-x(r1)))THEN kap(ip)=kap(ip)
(w(ip,r1)*(x(r1)-x(r1))):GOTO wei4
201 GP LOCATE 23,1:PRINT" Achtung, "+sp$(ip)+" !"
202 5n PRINT" Verlust Ihrer "+Ak$(r1)+"-Aktien!"
203 Av kap(ip)=kap(ip)+w(ip,r1)*x(r1):v(r1)=v(r1)+w(ip,r1):w(ip
,r1)=0
204 Hc1 wei4:
205 782 NEXT ip:x(r1)=z(r1):PRINT:PRINT" Automatische Durchföhrun
g!"
206 Kx hp=hp-3:GOSUB tast:GOTO kk
207 tJ0 'Hausse/Baisse
208 km HaBa:
209 YM1 COLOR 5,2:CLS
210 YW IF hp<3 THEN GOTO Notg
211 fN IF r1=2 OR r1=7 THEN up=1:dw=1 ELSE IF r1=3 OR r1=8 THEN u
p=1:dw=2
212 Jt IF r1=4 OR r1=9 THEN up=2:dw=1 ELSE up=2:dw=2
213 C1 r2=INT(RND*13):IF r2=0 OR r2=12 THEN GOSUB Wahl
214 h7 IF r1>6 THEN go3
215 23 IF r2<>0 AND r2<>12 THEN go2
216 77 go:
217 702 LOCATE 2,1:INPUT" AG-Nummer Hausse";r2:IF r2<1 OR r2>1
3 THEN go
218 tL1 go2
219 wL2 CLS:PRINT:PRINT" Hausse ":PRINT" -----"
220 vL PRINT" AG Diff. Alt Neu" PRINT
221 bH r1=INT(RND*5)+1
222 2C PRINT TAB(2)Ak$(r2)TAB(18)r1*up TAB(26)x(r2)TAB(33)x(r2)+
r1*up
223 WH PRINT" -----"
224 xZ PRINT:PRINT" Baisse " PRINT" -----"
225 OQ PRINT" AG Diff. Alt Neu" PRINT
226 dT FOR ip=1 TO 13:IF ip=r2 THEN go2a
227 ZB3 LOCATE 12+ip,2
228 aT PRINT Ak$(ip)TAB(18)-r1*dw TAB(26)x(ip)TAB(33)x(ip)-r1*d
w
229 o1 go2a
230 J2 NEXT ip:PRINT" -----"
231 hP LOCATE 27,2:INPUT" Akzeptieren Sie";jn$:IF jn$="n" OR jn$=
"N" THEN hp=hp-3:GOTO ak
232 nd FOR ip=1 TO 13
233 J3 IF ip=r2 THEN x(ip)=x(ip)+r1*up ELSE x(ip)=x(ip)-r1*dw
234 Ya2 NEXT ip:hp=hp-1:GOTO kk
235 EN1 go3:
236 d2 IF r2<>0 AND r2<>12 THEN go5
237 KU1 go4:
238 OP2 LOCATE 2,1:INPUT" AG-Nummer Baisse";r2:IF r2<1 OR r2>1
3 THEN go4
239 b1 go5
240 J2 CLS:PRINT:PRINT" Baisse " PRINT" -----"
241 Gg PRINT" AG Diff. Alt Neu":PRINT
242 wG r1=INT(RND*5)+1
243 9T PRINT TAB(2)Ak$(r2)TAB(18)-r1*dw TAB(26)x(r2)TAB(33)x(r2)
-r1*dw
244 ic PRINT" -----"
245 AU PRINT:PRINT" Hausse " PRINT" -----"
246 L1 PRINT" AG Diff. Alt Neu":PRINT
247 yP FOR ip=1 TO 13:IF ip=r2 THEN go6
248 wJ3 LOCATE 12+ip,2
249 3K PRINT Ak$(ip)TAB(18)r1*up TAB(26)x(ip)TAB(33)x(ip)+r1*up
250 fr1 go6:
251 g12 NEXT ip:PRINT" -----"
252 1A LOCATE 27,2:INPUT" Akzeptieren Sie";jn$:IF jn$="n" OR jn$=
"N" THEN hp=hp-3:GOTO kk
253 Bc FOR ip=1 TO 13
254 mP3 IF ip=r2 THEN x(ip)=x(ip)-r1*dw ELSE x(ip)=x(ip)+r1*up
255 tw2 NEXT ip:hp=hp-1:GOTO kk
256 oJ0 'Multiple Action Flood
257 wH Flood:
258 5p1 COLOR 3,0:CLS
259 A3 PRINT:PRINT" Multiple Action Flood"
260 Cn PRINT:PRINT" -----"
261 yW PRINT" AG Diff. Alt Neu"
262 O9 FOR ip=1 TO 13:r1=INT(RND*151)-75
263 Ig2 LOCATE 6+ip,2:PRINT Ak$(ip)TAB(18)r1;"%TAB(27)x(ip)TAB(3
4 x ip)+INT x ip,*r1,100)
264 Qu x(ip)=x(ip)+INT(x(ip)*r1/100):FOR t=1 TO 3000:NEXT t
265 4g1 NEXT ip:PRINT" -----"
266 3Z PRINT:PRINT" Automatische Durchführung!":hp=hp-3:GOSUB tas
t:GOTO kk
267 Zm0 'Manual 1
268 7a Ma1:
269 b81 COLOR 1,14:CLS
270 a2 PRINT:PRINT" Brokers Manual Activity I":GOSUB Wanl
271 Ya so:
272 Jk LOCATE 22,1:INPUT" Welche AG-Nummer soll steigen";r1
273 In IF r1<1 OR r1>13 THEN so
274 BV so2:
275 yY LOCATE 23,1:INPUT" Welche AG-Nummer soll fallen ";r2
276 zn IF r2<1 OR r2>13 OR r1=r2 THEN so2
277 JG COLOR 1,14:CLS
278 uX PRINT:PRINT" Brokers Manual Activity I":r3=INT(RND*50)+1
279 LP PRINT:PRINT" Es steigt"
280 NK PRINT" -----"
281 AW PRINT" AG Diff. Alt Neu":PRINT
282 HF PRINT TAB(2)Ak$(r1)TAB(18)r3;"%TAB(27)x(r1)TAB(34)INT(x,r

```



```

1)+x(r1)*r3/100)
283 Sd x r1 =INT x r1 +x r1 +r3 .00
284 RO PRINT "-----"
285 F2 PRINT PRINT " Es fä!t:"
286 TQ PRINT "-----"
287 Ge PRINT " AG Diff. Alt Neu":PRINT
288 na PRINT TAB(2)Ak$(r2)TAB(18)-r3; "%TAB(27)x(r2)TAB(34)INT(x(
r2)-x(r2)*r3/100)
289 s3 x(r2)=INT(x(r2)-x,r2)*.3/100,
290 XU PRINT "-----"
291 A5 PRINT:PRINT:PRINT " Automatische Durchführung!":hp=hp-3:GOS
UB tast:GOTO kk
292 3HO Manual 2
293 s4 Ma2.
294 OX1 COLOR 1,14:CLS
295 ay PRINT:PRINT " Brokers Manual Activity I!"
296 2t PRINT "-----":PRINT
297 26 PRINT " <1> Dividende / Aufwertung":PRINT " <2> Hausse
/Baisse (1)> (1/2)<"
298 JW PRINT " <3> Hausse/Baisse (1)> (1/2)<":PRINT " <4>
Hausse/Baisse (1)>> (1/2)<"
299 Nu PRINT " <5> Hausse/Baisse (1)>> (1/2)<":PRINT " <6>
> Multiple Action Flood"
300 bK PRINT " <7> Hausse/Baisse (1/2)> (1)<":PRINT " <8>
Hausse/Baisse (1/2)> (1)<:"
301 nh PRINT " <9> Hausse/Baisse (1/2)>> (1)<":PRINT " <10>
Hausse/Baisse (1/2)>> (1)<:"
302 7o PRINT " <11> Manual Activity I".PRINT " <12> Finanzamt"
303 Gy LOCATE 22,1:PRINT " Erklärung: >= normal steigend"
304 Kn PRINT " >>= stark steigend":PRINT "
<= normal fallend"
305 nn PRINT " <=<= stark fallend"
306 Ft a1
307 Jv2 LOCATE 18,1:INPUT " Börsenaktion-Nummer";r1:IF r1<1 OR r1
>12 THEN a1
308 nS ON r1 GOTO Div1,HaBa,HaBa,HaBa,HaBa,Flood,HaBa,HaBa,HaBa,
HaBa,Mai,Fiskus
309 Ta0 'Fiskus
310 Kv Fiskus
311 q81 max=0 maxi=0
312 01 COLOR 4,0:CLS
313 f1 PRINT PRINT " Der Fiskus bittet zur Kasse:".PRINT
314 00 PRINT " Kapitalertragssteuer"
315 uy PRINT "-----":PRINT
316 w5 PRINT " (25% des Barkapitals, sofern dieses "
317 A0 PRINT " größer ist als der Freibetrag)"
318 rg PRINT PRINT " Startkapital: ";kapital; "DM"
319 g0 PRINT " Freibetrag: ";INT(kapital*.15); "DM"
320 na PRINT PRINT:PRINT " Name Bar DM Stnetto Steuer"
:PRINT
321 sr FOR ip=1 TO s:LOCATE 15+ip,2
322 wC2 IF kap(ip)>INT(kapital*.15) THEN max=kap(ip)-INT(kapitel
*.15):maxi=max/4
323 2R PRINT sp$(ip)TAB(14)kap(ip)TAB(23)INT(max)TAB(33)INT(maxi
324 J3 IF kap(ip)>INT(kapital*.15) THEN kap(ip)=kap(ip)-INT(max
1):max=0:maxi=0
325 LT1 NEXT ip:PRINT:PRINT
326 oG PRINT " Automatische Durchführung!":GOSUB tast:GOTO k2
327 110 'MenÜpunkt Börsenkurse
328 Iz Kurse:
329 oF1 WINDOW 4, "Börsenkurse", (0,10)-(360,175), 0,2:COLOR 7,1:CLS
330 KW PRINT:PRINT " AG EKurs Kurs Stück"
331 Wn PRINT "-----"
332 gg FOR i=1 TO 13:LOCATE 3+i,2:PRINT Ak$(1)TAB(20)z(1)TAB(30)x
(1)TAB(39)v(1): NEXT 1
333 s4 PRINT:PRINT:PRINT " Bitte Taste drücken":GOTO kkk
334 ak0 MenÜpunkt Charts
335 tK Charts
336 4o1 WINDOW 4, "Char's", (0,10)-(630,240), 0,2
337 Je kre(p)=kre(p)+60
338 Y2 ausw
339 K22 COLOR 0,1:CLS
340 hR LOCATE 3,2:PRINT "Bitte wählen Sie:"
341 oI FOR i=1 TO 13 LOCATE 4+i,2
342 w43 PRINT Ak$(i)TAB(18) "<";i;">"
343 H12 NEXT i:LOCATE 24,1
344 ck PRINT " Zurück zum Menü: bei 1. < 0 > eingeben"
345 Xd PRINT " Einzeldarstellung: bei 2. < 0 > eingeben"
346 eb1 eingl.
347 Gg2 LOCATE 20,2:INPUT "1. AG-Nummer (0-13) ";gr(1):IF gr(1)<0
OR gr(1)>13 THEN eing
348 sd IF gr(1)=0 THEN kk
349 jgl eing2

```

```

350 9w2 LOCATE 21,2:INPUT "2. AG-Nummer (0-13) ";gr(2):IF gr(2)<0
OR gr(2)>13 THEN eing2
351 8J IF gr(2)=0 THEN ee=1:gr(2)=22 ELSE ee=2
352 T2 COLOR 1,0:CLS:COLOR 4,0:PRINT TAB (2) "1.Aktie: "+Ak$(gr(1
))
353 IL IF gr(2)<>22 THEN COLOR 3,0:PRINT TAB(2) "2.Aktie: "+Ak$(
gr(2))
354 sN COLOR 1,0:d=4
355 t0 LOCATE 5,1:PRINT " Over":PRINT " 2000":PRINT " 1900":PRINT "
1800":PRINT " 1700":PRINT " 1600"
356 2W PRINT " 1500":PRINT " 1400":PRINT " 1300":PRINT " 1200":PRINT
" 1100":PRINT " 1000":PRINT " 900"
357 LU PRINT " 800":PRINT " 700":PRINT " 600":PRINT " 500":PRINT
" 400":PRINT " 300":PRINT " 200"
358 g4 "414" 1000:PRINT " Flow":LINE(45,30)-(45,210)
359 dM LOCATE 27,7:PRINT "1 2 3 4 5 6 7 8 9 10 11 12 13 1
4 15 16 17 18 19 20 21 22"
360 JU LINE(15,207)-(600,207)
361 00 FOR ip=1 TO ee
362 ut3 iln=0:iln=0:ilv=0:ilb=0:ila=0:ile=0
363 40 iln=VAL MIDS a=1 gr 'p ' ' ' ilv=VAL MIDS(stS gr 'p.),
9,4))
364 86 ilv=207-(iln/12):ilb=207-(iln/12)
365 01 LINE 45,ilv - c9,ilb ,:
366 6a2 eig3
367 JY3 FOR i=3 TO 22
368 YD4 iln=VAL(MIDS(st$(gr(ip)),ic*4+1,4)):IF ila<20 THEN ein
g4
369 XM ilc=207- ila 12)
370 JX LINE -(ic*24+21,ile),d
371 Fm3 NEXT i0
372 I71 eig3
373 ht3 d=d-1
374 Lt2 NEXT ip
375 92 LOCATE 1,30:PRINT "Bitte Taste drücken":GOSUB tast2:GOTO
ausw
376 rV0 'MenÜpunkt Aktienkauf
377 LN Aktienkauf.
378 UC1 WINDOW 4, "Aktienkauf", (0,10)-(425,187), 0,2:COLOR 13,1:CLS
379 EB IF hp<2 THEN GOTO Notg
380 YB PRINT:PRINT " Name: "+sp$(p)+" Barkapital: ";kap(p); "DM"
381 OF PRINT:PRINT " Welche Aktie möchten Sie kaufen? <0
> Keine"
382 KL PRINT:PRINT "-----"
383 Bu PRINT " Nr. AG Vorrätig Kurs"
384 X8 FOR i=1 TO 13
385 Xe2 LOCATE 8+i,2:PRINT TAB(3) "<";i;">TAB(14)Ak$(1)TAB(34)v
(1)TAB(47)x 1
386 fv1 NEXT 1
387 86 ein
388 D12 LOCATE 4,33:INPUT anr:IF anr=0 THEN GOTO ein3 ELSE IF(anr
<0 OR anr>13)THEN ein
389 I3 COLOR 13,1:CLS
390 gY PRINT:PRINT " Aktiennummer " anr PRINT:PRINT "-----"
391 1z PRINT " AG Brief Kurs Vorrätig"
392 HL LOCATE 7,2:PRINT Ak$(anr)TAB(18)z(anr)TAB(28)x(anr)TAB(37
)v(anr)
393 dK PRINT "-----"
394 1L mini=v(anr):maxi=INT(kap(p)/x(anr))
395 4o IF mini>maxi THEN maxi=maxi ELSE maxi=mini
396 aU PRINT:PRINT " Maximale Ankaufsmenge in Stück: ";maxim
397 Us PRINT:INPUT " Ankaufsmenge in Stück, akte
398 JK IF akte>v(anr) THEN note$(anr)="BG" ELSE IF akte<1 THEN
note$(anr)="BBR" ELSE GOTO ein2
399 1h PRINT:PRINT " Ankaufsmenge zu klein!":GOSUB tast:GOTO Aktk
auf
400 DN1 ein2
401 Gx2 IF akte*x(anr)>kap(p)THEN PRINT:PRINT " Ihr Kapital reich
t nicht aus!":GOSUB tast:GOTO Aktienkauf
402 r7 an=anr:bn=x(an)
403 tX PRINT:PRINT:PRINT " Kaufpreis: ";akte*bn; "DM"
404 Ik PRINT " Maklergebühr: ";INT(akte*x(an)/50); "DM"
405 Er PRINT "-----":PRINT " Effektivsumme: ";
akte*bn+INT(akte*x(an)/50); "DM"
406 iG w(p,an)=w(p,an)+akte:note$(an)="GELD":kap(p)=kap(p)-akte*
bn:v(an)=v(an)-akte
407 79 IF kap(p)<0 THEN kre(p)=kre(p)+ABS(kap(p)):kap(p)=0
408 kT ku(p,an)=x(an)
409 4q PRINT:PRINT " Anweisung ausgeführt! Bitte Taste drücken"
410 U1 hp=hp-2:kre(p)=kre(p)+INT((akte*x(an))/50):GOTO kkk
411 Te1 ein3
412 Na2 GOTO kk

```

Listing 1. (Fortsetzung)

```

413 d10 'Menüpunkt Aktienverkauf
414 ML Aktverk.
415 401 WINDOW 4, "Aktienverkauf", (0,10)-(425,187), 0,2:COLOR 3,1:CLS
416 pm IF hp<2 THEN GOTO Notg
417 9m PRINT:PRINT "Name: ";sp$(p); "Barkapital: ";kap(p); "DM"
418 38 PRINT PRINT "Welche Aktie möchten Sie verkaufen? <
    0> Keine"
419 Lw PRINT PRINT "-----"
420 oT PRINT "Nr. AG Besitz Kurs"
421 8j FOR i=1 TO 13
422 a22 LOCATE 8+1,2:PRINT TAB(3)"<";1;">TAB(14)Ak$(1)TAB(34)v
    (p,1)TAB(47 x,1)
423 Gw1 NEXT i
424 ne egl:
425 s22 LOCATE 4,36:INPUT anr:IF anr=0 THEN GOTO ein3 ELSE IF(anr
    <0 OR anr>13)THEN egl
426 vs IF v(p,anr)=0 OR notes$(anr)="GELD" THEN notv ELSE GOTO ja
    ver
427 6n1 notv:
428 a22 PRINT "Verkauf nicht möglich! Bitte Taste drücken":GOSUB
    tast2:GOTO kk
429 LD1 Javer:
430 wh2 COLOR 3,1:CLS
431 LD PRINT:PRINT "Aktiennummer ",anr:PRINT:PRINT "-----"
432 xs PRINT "AG Brief Kurs Besitz"
433 0x LOCATE 7,2:PRINT Ak$(anr)TAB(18)s(anr)TAB(28)x(anr)TAB(37
    )v(p,anr)
434 Ix PRINT "-----"
435 MM PRINT:PRINT "Kauf der Aktie bei: ";ka(p,anr); "für insgesam
    t: ";ka(p,anr)*v(p,anr); "DM"
436 p4 PRINT:PRINT "Effektivkaufpreis pro Aktie: ";ka(p,anr)+INT(
    x(anr)/50); "DM"
437 Pu PRINT:PRINT:INPUT "Verkaufsmenge in Stück";aktv
438 wZ IF aktv<1 OR aktv>v(p,anr)THEN PRINT:PRINT "Verkaufsmen
    ge zu klein/groß!":GOSUB tast:GOTO Aktverk
439 xf IF aktv*x(anr)<1 THEN PRINT:PRINT "Totales Verlustgeschä
    ft !!!":GOSUB tast:GOTO Aktverk
440 xt an=anr v(p,an)=v(p,an)-aktv:notes$(an)="BRIEF":kap(p)=kap(
    p)+aktv*x(an):v(an)=v(an)+aktv
441 o7 PRINT PRINT "Verkaufswert ";x(an)*aktv; "DM" PRINT "Makle
    rgebühr: ";INT((x(an)*aktv)/50); "DM"
442 pm kre(p)=kre(p)+INT((x(an)*aktv)/50):PRINT:PRINT "Anweisun
    g ausgeführt! Bitte Taste drücken"
443 6i hp=hp-2:GOTO kkk
444 3J0 'Menüpunkt Kapitalliste
445 U9 Kaplist:
446 QP1 WINDOW 4, "Kapitalliste", (0,10)-(200,165), 0,2:COLOR 12,5:CLS
447 AL GOSUB Aktka
448 Mh PRINT:PRINT "Name: ";sp$(p)
449 xu PRINT PRINT "Bargeld: ";kap(p); "DM"
450 VE PRINT PRINT "Aktienwert ",Ak(p); "DM"
451 H3 PRINT:PRINT "-----":PRINT "BRUTTO: ";
    kap(p)+Ak(p); "DM"
452 10 PRINT:PRINT "Schulden: ";kre(p); "DM"
453 6R PRINT:PRINT "-----":PRINT "NETTO: ";
    kap(p)+Ak(p)-kre(p); "DM"
454 p1 PRINT:PRINT:PRINT "Bitte Taste drücken":GOTO kkk
455 1v0 'Menüpunkt Kapitalvergleich
456 ZH Kapver:
457 wJ1 IF no=0 THEN kre(p)=kre(p)+30:g$="Kapitalvergleiche"
458 j6 misp:
459 p22 IF s=2 THEN sp2 ELSE IF s=3 THEN sp3 ELSE sp4
460 EZ1 sp2:
461 7W2 WINDOW 4,g$(0,10)-(580,90), 0,2:COLOR 5,12:CLS
462 mS GOTO beve
463 Lh1 sp3:
464 Bu2 WINDOW 4,g$(0,10)-(580,107), 0,2:COLOR 5,12:CLS
465 pV GOTO beve
466 Sp1 sp4:
467 Cv2 WINDOW 4,g$(0,10)-(580,125), 0,2:COLOR 5,12:CLS
468 1j beve:
469 4N3 PRINT:PRINT "Name Bargeld Aktienwert
    Brutto Schulden Netto"
470 1c PRINT "-----"
471 Yj GOSUB Aktka
472 Od4 FOR ip=1 TO s
473 7v5 PRINT TAB(2)sp$(ip)TAB(20)kap(ip)TAB(31)Ak(ip)TAB(44)k
    ap(ip)+Ak(ip)TAB(53)kre(ip)TAB(65)kap(ip)+Ak(ip)-kre(ip)

```

```

474 h64 PRINT NEXT ip
475 eu3 PRINT:PRINT "Bitte Taste drücken":no=0
476 Fr GOTO kkk
477 n30 'Menüpunkt Aktienliste
478 RA Aktlist.
479 eX1 WINDOW 4, "Aktienliste", (0,10)-(420,235), 0,2:COLOR 2,0:CLS
480 bs GOSUB Aktka
481 tK PRINT:PRINT "Name: ";sp$(p)
482 PM PRINT:PRINT "Unverbrieftes Kapital: ";kap(p); "DM"
483 TN PRINT "Verbrieftes Kapital ";Ak(p); "DM"
484 Is PRINT "-----"
485 po PRINT "AG Ums. Habe Kurs":PRINT
486 Qa FOR i=1 TO 13:LOCATE 8+1,2:PRINT Ak$(i)TAB(22)notes$(i)TAB(
    31 v(p,1)TAB(40 x,1)
487 ps NEXT i:PRINT "-----"
488 r0 PRINT:PRINT "Gesamtkapital: ";kap(p)+Ak(p); "DM"
489 Oa PRINT:PRINT:PRINT "Bitte Taste drücken":GOTO kkk
490 Ik0 'Menüpunkt Aktienvergleich
491 YK Aktver:
492 2J1 kre(p)=kre(p)+40
493 IR IF s=2 THEN s2 ELSE IF s=3 THEN s3 ELSE s4
494 CF s2:
495 Gv2 WINDOW 4, "Aktienvergleich", (0,10)-(360,170), 0,2:COLOR 0,2
    CLS
496 oo PRINT PRINT "AG"TAB(21)sp$(1)TAB(35)sp$(2)
497 CT PRINT "-----"
498 w6 FOR ip=1 TO 13:LOCATE 3+ip,2
499 1a3 PRINT Ak$(ip)TAB(20)v(1,ip)TAB(34)v(2,ip)
500 QV2 NEXT ip:PRINT:PRINT:PRINT "Bitte Taste drücken":GOTO kkk
501 WQ1 s3:
502 S22 WINDOW 4, "Aktienvergleich", (0,10)-(470,170), 0,2:COLOR 0,2
    CLS
503 kt PRINT PRINT "AG"TAB(21)sp$(1)TAB(35)sp$(2)TAB(48)sp$(3)
504 Gs PRINT "-----"
505 3D FOR ip=1 TO 13:LOCATE 3+ip,2
506 e13 PRINT Ak$(ip)TAB(20)v(1,ip)TAB(34)v(2,ip)TAB(47)v(3,ip)
507 Xa2 NEXT ip:PRINT:PRINT:PRINT "Bitte Taste drücken":GOTO kkk
508 Wb1 s4:
509 bK2 WINDOW 4, "Aktienvergleich", (0,10)-(370,170), 0,2:COLOR 0,2
    CLS
510 5p PRINT:PRINT "AG"TAB(21)sp$(1)TAB(35)sp$(2)TAB(48)sp$(3)TA
    B(61)sp$(4)
511 qK PRINT "-----"
512 AK FOR ip=1 TO 13:LOCATE 3+ip,2
513 xP3 PRINT Ak$(ip)TAB(20)v(1,ip)TAB(34)v(2,ip)TAB(47)v(3,ip)T
    AB(60)v(4,ip)
514 aj2 NEXT ip:PRINT:PRINT:PRINT "Bitte Taste drücken":GOTO kkk
515 3R0 'Menüpunkt Nächster
516 7i Nichte:
517 rR1 CALL requester("Spielerwechsel", "Ja", "Nein")
518 yz LOCATE 18,2
519 xy IF req=2 THEN WINDOW OUTPUT 2:req=0:GOTO c
520 Z8 IF req=1 THEN wech
521 20 GOTO Nichte
522 DI wech:
523 fa2 WINDOW 4, "Spielerwechsel", (0,10)-(330,240), 0,2:COLOR 1,5:
    CLS
524 L4 PRINT:PRINT "Bitte warten Sie: AMIGA rechnet!":hp=5
525 kP IF kre(p)>0 AND kap(p)>kre(p) THEN GOSUB Schulden
526 Au IF kre(p)>0 AND kap(p)<kre(p) THEN GOSUB Zinsen
527 Tr GOSUB Aktka:krelim=kapital/2
528 rS FOR i=1 TO 13
529 AN3 IF x(i)<50 THEN GOSUB Low ELSE IF x(i)>2000 THEN GOSUB
    High
530 su h$=STR$(x(i)):IF LEN(h$)=4 THEN h$=" "+h$ ELSE IF LEN(h$
    )=3 THEN h$=" "+h$
531 nt st$(i)=MID$(st$(i),5,88)+MID$(h$,2,4)
532 PU IF notes$(i)="GELD" THEN notes$(i)="BZ"
533 212 NEXT i
534 OX IF (kap(p)+Ak(p))>=spielende AND kre(p)=0 THEN endes=1
535 FH IF ((kap(p)+Ak(p)-kre(p)<100)OR kre(p)>krelim)THEN ende
    s=1
536 eo p=p+1:IF p>s THEN p=1
537 ly IF endes THEN Spielend
538 h1 PRINT:PRINT PRINT "Nächster Spieler ist: ";sp$(p)
539 GO PRINT:PRINT "Bitte Taste drücken":GOTO kkk
540 Lj1 Spielend
541 jX2 GOSUB tast:WINDOW CLOSE 4:WINDOW OUTPUT 2
542 gh g$="Schlubberechnung":CLS:GOTO misp
543 BNO 'Menüpunkt Übersicht
544 W8 Uebers:

```



```

545 gw1 WINDOW 4,"Übersicht",(0,10)-(250,140),0,2:COLOR 5,6:CLS
546 w0 PRINT FOR i=1 TO s:PRINT "Spieler";i;": ";sp$(i):PRINT:N
EXT i
547 h4 PRINT:COLOR 6,5:LOCATE ,2:PRINT "Startkapital :";kapital;
DM"
548 w1 LOCATE ,2:PRINT "Spielende bei: ";spielende;"DM":COLOR 5,6:P
RINT
549 IW GOSUB tast:GOTO kk
550 5t0 'Menüpunkt Help
551 Oz Help:
552 fd1 WINDOW 4,"BROKER - Das Börsenspiel von Sebastian Peetz",(0
,10)-(630,200),0,2
553 TO COLOR 5,15:CLS
554 Iv PRINT
555 8S PRINT " Menüpunkt                                Erklärung
DM HP"
556 rT PRINT "-----"
557 Fx PRINT "Laden                                Ein Spiel wird von Disk gel
aden                                ]"
558 Bv PRINT "Speichern                                Ein Spiel wird abgespeicher
t                                ]"
559 zP PRINT "Neues Spiel                                Es wird ein neues Spiel beg
onnen                                ]"
560 xH PRINT "Quit                                Spielende ohne die Spielstä
nde abzuspeichern                                ]"
561 41 PRINT "Börsenaktion                                Der Spieler kann aktiv die
Börsenkurse                                20] 1-3"
562 uS PRINT "                                durch geschicktes Handeln b
eeinflussen                                ]"
563 Mr PRINT "Börsenkurse                                Es werden die aktuellen Bör
senkurse angezeigt                                - ] -"
564 VJ PRINT "Charts                                Aktienkurse können graphise
h dargestellt werden                                60 -"
565 WJ PRINT "Aktienkauf                                Aktien können zum Tageskurs
gekauft werden                                5% 2"
566 H1 PRINT "Aktienverk.                                Aktien können zum Tageskurs
verkauft werden                                5% 2"
567 3G PRINT "Kapitalliste                                Eigenes Kapital (Aktienkapi
tal und bar)                                - , -"
568 Iv PRINT "Kapitalvergl.                                Kapital aller Spieler
30] -"
569 m6 PRINT "Aktienliste                                Auflistung der eigenen Akti
en (Kurse,Stück)                                - ] -"
570 w2 PRINT "Aktienvergl.                                Aktien aller Spieler (Stück
)                                40] -"
571 aJ PRINT "Nächster                                Der nächste Spieler ist an
der Reihe                                ]"
572 XQ PRINT "Übersicht                                Alle wichtigen Daten zum Sp
iel im Überblick                                ]"
573 IT PRINT "Info                                Diese Seite
574 rb PRINT:PRINT "Bitte Taste drücken":GOTO kkk
575 FPO 'Unterprogramme
576 jf 'Subprogramme für Menüpunkt Börsenaktion/Hausse-Baisse
577 WA Wahl:
578 hw1 LOCATE 4,1:PRINT "-----"
579 bn PRINT " Nr.      AG              Habe      Kurs"
580 h1 FOR i=1 TO 13
581 x12 LOCATE 6+1,2 PRINT "<";i;"> TAB(11)Ak$(i)TAB(28)w(p,i)T
AB(36)x(i)
582 391 NEXT i:PRINT "-----"
583 1N RETURN
584 h20 Subprogramme für Menüpunkt Nächster
585 qG Schulden.
586 WR1 PRINT:PRINT:PRINT "Schulden:" PRINT "-----"
587 99 PRINT PRINT " "+sp$(p)+" mLB ";kre(p),"DM bezahlen."
588 EN kap(p)=kap(p)-kre(p) kre(p)=0
589 rT RETURN
590 620 Zinsen
591 t01 PRINT:PRINT:PRINT "Schulden: ";kre(p);"DM":PRINT"+ Zinsen
";CINT(kre(p)/10);"DM"
592 K1 PRINT "-----":PRINT:PRINT "Gesamt: ";kre(p
)+CINT(kre(p)/10);"DM"
593 Io kre(p)=kre(p)+CINT(kre(p)/10)
594 wY RETURN
595 YMO Low'
596 QJ1 PRINT:PRINT:PRINT "Aufwertung:":PRINT "-----".PRINT
597 ga PRINT "Kurswert zu niedrig bei: "+Ak$(i)+" !"
598 mY PRINT "Aufwertung je Aktie :";(z(1)-x(i));"DM" PRINT
599 Rg FOR ip=1 TO s
600 9I2 IF kap(ip)<w(ip,i)*(z(1)-x(i)) THEN verl
601 9n PRINT TAB(2)sp$(ip)TAB(13)"mLB TAB(17)w(ip,i)*(z(1)-x(i))

```

```

:DM zahlen."
602 8D kap(ip)=kap(ip) w(ip,i)*(z(1)-x(i))
603 M0 GOTO weiter
604 nA1 verl:
605 sk3 PRINT TAB(2)sp$(ip)TAB(13)"- AKTIENVERLUST -"
606 -X kap(ip)=kap(ip)+w(ip,i)*x(i)-v(1)=v(1)+w(ip,i) w(ip,i)=0
607 0L1 weiter:
608 7f2 NEXT ip
609 Iv x(i)=z(i)
610 07 PRINT:PRINT" - TASTE -" GOSUB tast2
611 3p1 RETURN
612 530 tige:
613 3e1 PRINT:PRINT:PRINT "Dividende:":PRINT"-----".PRINT
614 pP PRINT "Kurswert zu hoch bei "+Ak$(i)+" !"
615 7S PRINT "Auszahlung je Aktie : ";(x(1)-2000);"DM":PRINT
616 ix FOR ip=1 TO s
617 0K2 PRINT TAB(2)sp$(ip)TAB(13)"erhält TAB(20);(w(ip,i)*(x(1)-
2000));"DM"
618 -y kap(ip)=kap(ip)+w(ip,i)*(x(1)-2000))
619 1q1 NEXT ip
620 9D x(1)=2000
621 BI PRINT:PRINT" - TASTE -":GOSUB tast2
622 00 RETURN
623 k50 Sonstige Subprogramme
624 M Aktea
625 r61 FOR ip=1 TO s
626 fQ2 Ak(ip)=0
627 S33 FOR i=1 TO 13
628 h4 Ak ip)=Ak(ip)+x(i)*w(ip,i)
629 aq3 NEXT i
630 w21 NEXT ip:RETURN
631 9z0 Notg:
632 hM1 PRINT:PRINT:PRINT "Sie haben nicht genug Handlungspunkte!"
633 mW PRINT:PRINT "Bitte Taste drücken":GOTO kkk
634 yb0 Hpu:
635 7f1 LOCATE 11,24:PRINT "Spieler: ";sp$(p)
636 3Y LINE(175,90)-(460,125),3,BF
637 5U LOCATE 14,23:COLOR 8,3
638 6V PRINT "Sie haben noch";hp;"Handlungspunkte! "
639 4M COLOR 1,0
640 gI RETURN
641 kd0 NoHpu:
642 Bm1 LOCATE 11,24:PRINT "Spieler: ";sp$(p)
643 Go LINE(175,90)-(490,125),3,BF
644 Cb LOCATE 14,23:COLOR 8,3
645 EL PRINT "Sie haben keine Handlungspunkte mehr! "
646 BT COLOR 1,0
647 nP RETURN
648 Nm0 Fehler:
649 by1 WINDOW 5,"ERROR - ERROR - ERROR",{310,10)-(610,80),0,2:COL
OR 1,5:CLS
650 13 PRINT:PRINT" Es ist folgender Fehler aufgetreten:":PRINT
651 Wc IF ERR=53 THEN PRINT "DATEI NICHT GEFUNDEN !":GOTO dud1
652 4Y IF ERR=61 THEN PRINT "DISKETTE IST VOLL !":GOTO dud1
653 my IF ERR=64 THEN PRINT "FALSCHER DATEINAME !":GOTO dud1
654 iJ IF ERR=67 THEN PRINT "ZU VIELE DATEIEN !":GOTO dud1
655 Vh IF ERR=70 THEN PRINT "DISKETTE SCHREIBGESCHÜTZT !":GOTO du
d1
656 W1 IF ERR=74 THEN PRINT "UNBEKANNTE DISKETTE".GOTO dud1
657 Xx PRINT "UNBEKANNTER FEHLERTYP!"
658 im dud1:
659 QJ2 PRINT:GOSUB tast:WINDOW CLOSE 5:RESUME kk
660 SJO 'Häufig verwendete Sub-Programme
661 1y k:
662 Zk1 WINDOW CLOSE 4
663 CU WINDOW OUTPUT 2
664 Yo GOTO Kurse
665 he0 k2:
666 6o1 WINDOW CLOSE 4
667 GY WINDOW OUTPUT 2
668 VM no=1:GOTO Kapver
669 w40 z:
670 Aa1 WINDOW CLOSE 4
671 Kc WINDOW OUTPUT 2
672 gk GOTO c
673 10 kkk
674 3w1 IF INKEY$="" GOTO kkk
675 -x WINDOW CLOSE 4
676 Ph WINDOW OUTPUT 2
677 Oc IF endes THEN Ende
678 mq GOTO c
679 Ss0 tast:

```

Listing 1. (Fortsetzung)

```

680 2M1 PRINT PRINT "Laser Laser Drucker"
681 JJ WHILE INKEY$=""
682 RF WEND
683 N2 RETURN
684 bx0 test2:
685 mN1 WHILE INKEY$=""
686 VJ WEND
687 t3 GET EN
688 ty0 Diverse Daten
689 rR papiere
690 291 DATA Bauwerke,lexikon,Broschüren,Kosmetik,Extraktivität,Chemie
,Pharmazie
691 Is DATA Maschinentechnik,Ausstattung,Automobil,Banken,Versicherung
692 000 merua
693 3S1 DATA 5
694 Q DATA 4
695 r2 DATA 1,Broker
696 r4 DATA 1,"Innen"
697 rR DATA 1,"Spezial"
698 rQ DATA 1,"Neues Spiel"
699 6n DATA 1,"Quiz"
700 gT DATA 3
701 11 DATA 1,Börse
702 15 DATA 1,"Börsenaktion"
703 16 DATA 1,"Börsenliste"
704 3N DATA 1,"Charts"
705 JV DATA 2
706 16 DATA 1,Aktier
707 On DATA 1,"Aktienauf"
708 Vt DATA 1,"Aktienverkauf"
709 rf DATA 4
710 64 DATA 1,Kapital,Aktienwerte
711 FM DATA 1,Kapitalliste
712 qu DATA 1,Kapitalvergleich
713 Dt DATA 1,Aktienliste
714 FU DATA 1,Aktienvergleich

```

```

715 vi DATA 3
716 DATA 1,Information
717 rH DATA 1,"Neuerster"
718 r4 DATA 1,"Börsenliste"
719 14 DATA 1,"ne"
720 20 merua
721 141 DATA 5
722 48 DATA 4
723 41 DATA 1,"Foster"
724 19 DATA 1,"Foster"
725 4 DATA 1,"Spezial"
726 06 DATA 1,"Neues Spiel"
727 1F DATA 1,"Quiz"
728 1V DATA 3
729 41 DATA 1,Börse
730 28 DATA 1,"Börsenaktion"
731 1b DATA 1,"Börsenliste"
732 4c DATA 1,"Charts"
733 0x DATA 2
734 1b DATA 1,"Aktier"
735 57 DATA 1,"Aktienauf"
736 10 DATA 1,"Aktienverkauf"
737 37 DATA 4
738 4Z DATA 1,Kapital,Aktienwerte
739 11 DATA 1,Kapitalliste
740 8F DATA 1,Kapitalvergleich
741 40 DATA 1,Aktienliste
742 4Z DATA 1,Aktienvergleich
743 14 DATA 3
744 gD DATA 1,Information
745 rR DATA 1,"Neuerster"
746 14 DATA 1,"Börsenliste"
747 7C DATA 1,"ne"
748 48T

```

Listing 1. (Schluß)

Packen wir's

**Wollen Sie mehr Programme
auf einer Diskette unter-
bringen? Die Lösung heißt
»Master-
Cruncher«.
Er packt alles.**

A migaBasic mit 103484 oder mit 71412 Byte? Die Entscheidung fällt nicht schwer: die kürzere Version ist besser. Ein kompletter Aztec-C-Compiler inklusive Source-Debugger mit 263388 oder mit 156292 Byte? Auch hier ist das Ergebnis klar. Wie diese Beispiele deutlich machen, können Sie erstaunlich viel Speicherplatz auf Ihren Disketten einsparen, wenn Sie unser Programm des Monats besitzen. In der Regel handelt es sich um 30 bis 50 Prozent der Originallänge. Die Funktionsweise von »MasterCruncher« ist im Prinzip einfach. Programme und Dateien enthalten gleiche Bytefolgen. Wenn diese Sequenzen zu kürzeren zusammengefaßt werden, heißt dies »packen« oder »crunchen«. Dabei muß man zwischen drei Typen von Dateien unterscheiden:

- verschiebbare Programme,
- Daten,
- nicht verschiebbare Programme.

Im Normalfall gehören die Programme auf dem Amiga zur ersten Kategorie. Nur wenige Ausnahmen fallen in die dritte Gruppe. Der Unterschied liegt darin, daß bei den nicht verschiebbaren Programmen ein bestimmter Speicherbereich verwendet wird. Gepackte Programme lassen sich genau wie vorher von der Workbench oder dem CLI aufrufen. Zu diesem Zweck muß die gepackte Datei im Speicher zuerst entpackt werden. Das erledigt MasterCruncher automatisch. Vor die komprimierten Daten wird ein Programm geschrieben, das diese Aufgabe übernimmt. Bei langen Programmen verringert sich sogar die Ladezeit von Diskette, trotz des notwendigen Entpackens.

Bei Daten ist eine andere Vorgehensweise nötig. Da hier keine Programmverzweigung auf die Daten erfolgt, können sie nicht im

Speicher entpackt werden. Komprimierte Daten werden mit MasterCruncher geladen und anschließend ungepackt auf Diskette gespeichert. Dann können Sie wiederverwendet werden.

MasterCruncher liegt in Form eines DATA-Laders vor, damit ihn jeder Amiga-Besitzer abtippen kann. Das Amiga-Basic-Programm (siehe Listing) generiert auf der Diskette das lauffähige Programm. Vor dem Laden des Listings müssen Sie allerdings den Befehl »CLEAR 40000« eingeben. Der Aufruf

MasterCruncher

startet das Programm vom CLI aus. Vor der Benutzung von der Workbench aus muß noch ein passendes Programmsymbol (Icon) kopiert werden oder mit IconEd ein eigenes gezeichnet werden. Der Typ des Icons muß Tool sein. Dann kann man MasterCruncher durch Doppelklick auf das Programmsymbol starten.

Da MasterCruncher selbst gepackt wurde (statt 14036 nur noch 9328 Byte), flackert kurz der Bildschirm und dann erscheint das Bedienungsfeld (siehe Bild). Betrachten wir nun die normale Vorgehensweise.

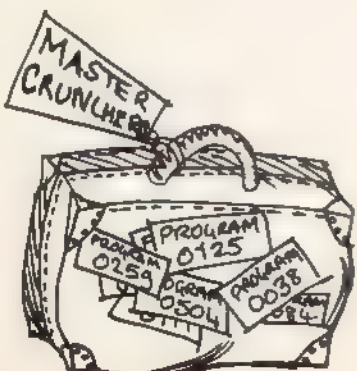
LOAD FILE/SAVE FILE

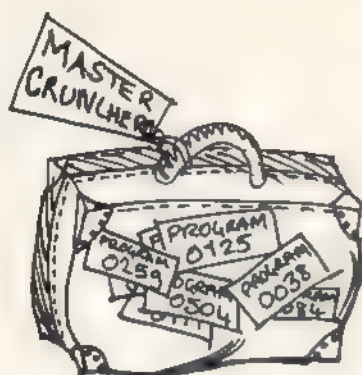
Zunächst laden Sie die zu packende Datei. Dazu klicken Sie einmal auf den Schalter (Gadget) »LOAD FILE«. Nach Betätigung erscheint ein Datei-Requester, mit dessen Hilfe Sie die Datei auswählen. Oberhalb der Einträge finden Sie ein Textfeld (Stringgadget) für den Pfad. Nach Anklicken erscheint ein Cursor zur Eingabe des neuen Pfads. Die Pfade »DF0:« und »DF1:« erreichen Sie durch Anklicken der entsprechenden Schalter. Unterhalb der Einträge sehen Sie ein Stringgadget, in dem der Name der Datei eingegeben

47 8d DATA 20,6d,00,0a,20,10,58,60,4a,7c
48 42 DATA 22,80,52,87,53,87,67,1a,22,50
49 63 DATA 20,28,00,04,e5,88,22,80,20,28
50 T3 DATA 00,00,00,00,00,00,00,00,00,00
51 16 DATA 51,80,00,02,20,1f,20,03,00,0a
52 d7 DATA 43,00,00,00,00,00,00,00,00,00
53 ae DATA 50,88,22,88,22,6d,00,0e,20,07
54 ae DATA e7,88,4e,ae,ff,2e,22,6d,00,18
55 MM DATA 11,89,00,11,4e,ae,ff,2e,ae,ff
56 qq DATA 71,ff,ae,ff,2e,ae,ff,2e,ae,ff
57 0a DATA 4a,9e,2f,1b,ef,00,00,00,00,00
58 0a DATA 22,6d,00,0a,22,06,e7,89,22,71
59 11 DATA 18,00,50,59,4e,ae,ff,2e,ae,ff
60 n1 DATA ff,3e,01,0f,00,00,00,00,00,00
61 m2 DATA 4a,85,67,02,52,86,7a,01,4e,75
62 xN DATA 4a,9b,20,6d,00,0e,20,06,e7,88
63 2E DATA 22,70,08,00,50,89,20,1b,67,24
64 qd DATA 22,1b,e7,89,20,0d,00,00,00,00
65 11 DATA 18,00,50,59,4e,ae,ff,2e,ae,ff
66 gM DATA 22,1b,e7,89,20,0d,00,00,00,00
67 vN DATA 51,89,00,11,4e,ae,ff,2e,ae,ff
68 lK DATA 48,e7,89,20,0d,00,00,00,00,00
69 vx DATA ff,ff,22,6d,00,00,00,00,00,00
70 Lo DATA 2b,49,00,14,2b,7c,00,07,00,00
71 a9 DATA 00,18,61,0c,4e,b9,00,00,00,00
72 00 DATA 00,00,00,00,00,00,00,00,00,00
73 00 DATA 2a,18,22,6d,00,18,d3,c5,41,80
74 n0 DATA 91,00,00,00,00,04,6a,60,6a,02
75 rK DATA 53,b8,10,20,0e,00,00,00,00,00
76 Y6 DATA e3,10,64,5a,22,4e,ae,ff,2e,ae,ff
77 qn DATA 10,20,e3,10,00,22,4e,ae,ff,2e,ae,ff
78 00 DATA 76,03,42,41,14,33,30,00,48,82
79 Xp DATA 78,ff,e5,6a,46,4a,53,2e,ae,ff,2e,ae,ff
80 jn DATA 66,04,10,20,e3,10,e3,ff,2e,ae,ff
81 Yt DATA 71,ff,4a,43,e7,2e,bf,01,5c,7e
82 p2 DATA ff,da,1a,03,33,7c,52,ae,ff,2e,ae,ff
83 of DATA 60,00,00,0a,0a,03,02,02,0e,07
84 xw DATA 04,01,13,20,33,c8,00,ff,2e,ae,ff
85 jA DATA 51,09,ff,ff,64,7a,0a,00,66,26,6b
86 vA DATA 00,14,50,8b,b1,eb,6f,00,00,0e,8
87 wT DATA 47,fa,00,3a,74,03,e3,08,66,04
88 Rb DATA 10,20,e3,10,6a,0a,5a,00,ff,2e,ae,ff
89 t9 DATA 42,41,52,42,16,33,20,00,67,12
90 QV DATA 48,83,53,43,e3,08,66,04,10,20
91 KH DATA e3,10,e3,51,51,eb,ff,ff,ff,ff,ff
92 05 DATA 20,05,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
93 05 DATA 0a,02,01,00,00,0a,06,04,03,02
94 wK DATA 0e,41,00,04,67,42,47,fa,00,3a
95 h5 DATA 76,01,e3,0b,6c,0a,1a,20,ae,ff,2e,ae,ff
96 04 DATA 64,0a,5a,0b,ff,ff,2e,ae,ff,2e,ae,ff
97 08 DATA 18,33,30,00,48,84,e3,08,66,04
98 aJ DATA 10,20,e3,10,e3,ff,2e,ae,ff,2e,ae,ff
99 08 DATA e3,4e,0a,ff,30,0a,00,00,00,00
100 99 DATA 0b,04,07,00,01,20,00,00,00,20
101 jf DATA 00,00,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
102 00 DATA 66,04,10,20,e3,10,e3,ff,2e,ae,ff
103 99 DATA 78,04,e3,08,66,04,10,20,e3,10
104 jA DATA e3,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
105 81 DATA 20,00,48,01,d5,c1,53,41,13,22
106 jz DATA 51,09,ff,ff,64,7a,0a,00,66,04,75
107 ge DATA 00,00,00,00,00,00,00,00,00,00
108 00 DATA 00,00,00,00,00,00,00,00,00,00
109 11 DATA 00,00,00,00,00,00,00,00,00,00

110 EC DATA 00,00,ae,0a,00,00,03,ea,00,00
111 53 DATA 08,38,00,00,36,e4,00,00,20,de
112 74 DATA b2,73,1a,cc,09,ae,00,ff,1a,ae
113 2Q DATA 49,33,66,e9,5a,11,0d,81,c4,54
114 x DATA b1,93,c9,ce,51,da,28,40,db,2d
115 58 DATA 1a,4a,07,00,ae,66,1a,00,00,3c
116 0E DATA 87,ec,00,5c,a4,e4,8c,a6,2e,02
117 11 DATA 4e,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
118 34 DATA 70,13,1a,01,fa,00,ae,2a,02,9c
119 5A DATA 47,50,fe,7a,89,9e,ff,2e,ae,ff
120 8Q DATA 8e,d4,0a,c0,db,82,25,25,e0,da
121 0C DATA f5,32,b0,ee,57,08,1c,0c,03,f1
122 p0 DATA d9,0a,2d,4d,91,e2,02,f6,e5,35
123 4C DATA ea,39,7b,2a,30,64,11,00,00,02
124 8C DATA 06,bc,5d,b7,28,00,3c,0a,d4,66
125 05 DATA fb,e9,29,14,20,3a,3c,2c,ec,87
126 11 DATA 06,4a,31,40,00,02,48,40,40,33
127 0a DATA 29,7a,6a,0e,dc,3c,3e,71,3a,31
128 13 DATA b0,03,00,ff,02,00,00,00,00,00
129 21 DATA 7d,da,fo,ab,2a,16,08,e4,97,be
130 14 DATA bb,05,88,14,ca,25,c2,bc,dc,9d
131 1A DATA df,2f,0c,b2,04,01,ae,0d,ce,0c
132 71 DATA ea,0a,10,0c,b2,11,64,0b,00,ff,ff
133 1E DATA d0,70,ff,42,31,ff,20,33,ae,7a
134 1E DATA 44,15,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
135 5P DATA 33,2f,ae,ff,2e,ae,ff,2e,ae,ff,2e,ae,ff
136 09 DATA 09,94,52,9d,7a,7b,0c,0a,ae,9a
137 3C DATA 7a,0b,07,29,fd,ae,ff,2e,ae,ff,2e,ae,ff
138 6C DATA 20,2e,8f,cd,30,72,ff,02,0e,4b
139 9a DATA 3a,02,fd,25,04,d7,0c,2c,7e,b2
140 70 DATA 1f,58,00,07,df,ae,1a,06,6b,eb
141 xx DATA 0c,30,97,ff,6b,2d,ae,ff,2e,ae,ff
142 1P DATA d9,4d,30,92,0c,ae,ff,2e,ae,ff,2e,ae,ff
143 7b DATA 25,10,c3,67,7b,5b,e6,6c,40,34
144 0a DATA 14,b0,51,80,01,94,e7,47,c2,99
145 PH DATA 19,04,30,2e,4b,50,6d,e7,61,ea
146 0a DATA 93,02,3f,d3,30,96,15,af,ff,2e,ae,ff
147 23 DATA 91,71,6c,ca,ff,b2,b2,05,30,82
148 21 DATA 9c,85,02,24,b0,3c,13,01,78,1e
149 6E DATA 30,6a,0c,06,45,6f,62,6b,32,e4
150 0d DATA 4e,0a,20,67,0a,70,5a,fe,3a,97
151 XT DATA 05,30,79,18,9a,6b,3a,6b,97,7c
152 a0 DATA 22,39,d4,fe,86,97,95,70,b0,8d
153 8R DATA ba,84,73,dd,a4,de,6e,00,78,20
154 1x DATA 50,20,68,00,0e,d1,fa,71,65,68
155 0a DATA 96,30,0c,22,90,00,07,78,fa,20
156 05 DATA 89,9b,2f,ff,2e,88,05,5d,30,02
157 86 DATA 0a,45,80,2a,ae,db,45,40,ae,72
158 01 DATA 23,0a,35,89,49,79,78,8c,0c,29
159 00 DATA 00,0c,00,8c,42,6a,30,88,20,1f
160 0E DATA 0d,7b,5b,43,fa,0a,96,2a,3e,6d
161 06 DATA 6c,23,e9,6e,00,8c,03,69,6e,74
162 0V DATA 7f,59,74,69,0f,6e,2e,6c,69,02
163 06 DATA 72,61,0e,00,71,3b,59,3a,23,fa
164 03 DATA 0b,bb,23,ae,6e,20,70,20,b8,51
165 1c DATA 1f,00,2f,9a,23,ee,21,27,71,4c
166 0V DATA 80,22,7a,2f,92,23,3c,5e,42,10
167 07 DATA fe,a6,7d,9d,75,b7,47,0e,ae,19
168 11 DATA 01,66,ff,b9,ff,9e,0f,6a,7c,fe
169 09 DATA e3,e1,7c,38,10,de,41,0a,3b,3a
170 09 DATA 00,00,0b,ae,af,2f,10,fe,61,1a
171 2f DATA da,43,9a,39,9b,fe,5a,49,00,7b
172 0K DATA 5a,44,0e,93,00,26,61,0e,76
173 86 DATA 05,95,7b,ab,b6,42,18,65,7e,98
174 00 DATA 23,58,42,83,24,08,52,43,4a,18
175 01 DATA 0c,1a,19,ae,ff,2e,ae,ff,2e,ae,ff
176 0Q DATA 20,0f,6e,0f,75,67,08,27,6d,0f
177 0q DATA 6d,6f,72,79,20,66,72,65,65,07
178 81 DATA fe,2f,92,0a,15,63,6e,4e,16,7d
179 87 DATA 6b,d9,12,26,73,5f,b4,9e,22,6d
180 18 DATA 32,25,37,9b,31,4b,01,8a,97,0c
181 1a DATA 86,4d,e6,b9,26,73,43,b6,96,ca
182 1c DATA 6d,33,cd,b7,9b,fa,9b,23,80,07
183 1a DATA 4a,06,31,19,97,3d,cd,e6,ae,6e
184 35 DATA 73,4d,00,9d,39,99,a0,5b,4d,4d
185 1Z DATA 64,3e,0f,63,b2,18,5a,6a,b6,cd
186 1a DATA 6d,7c,66,7a,2e,1c,76,36,03,00
187 01 DATA ea,00,03,d1,de,6e,0e,ae,94,6d
188 90 DATA c6,3a,c4,83,d4,1a,5d,c4,a0,e2
189 01 DATA 8a,7c,3a,bc,6c,0e,97,11,ca,01
190 01 DATA 14,71,b6,81,0b,78,4e,6d,0b
191 1a DATA 70,e5,6a,c1,02,3c,39,96,c9,2d

192 76 DATA 70,ff,9c,52,91,38,0a,ea,80,82
193 0D DATA 16,00,68,63,72,69,fe,ff,2e,ae,ff
194 0e DATA 31,2b,b7,ae,6c,ff,16,0d,bc,9f
195 01 DATA 2c,fd,10,2e,15,25,2c,58,97,1e
196 0A DATA 01,b4,0a,0a,0a,40,70,16,98,08,ae
197 01 DATA 5b,55,24,b0,0f,26,b4,ab,e4,34
198 0A DATA 3e,0e,b5,b4,c8,ae,28,73,44,03
199 11 DATA eb,01,80,07,2c,12,65,1e,d1,9c
200 01 DATA 0a,0a,52,9f,49,30,ae,9b,73,d8
201 0a DATA 52,ff,ec,b2,10,2c,7a,2a,80,d8
202 05 DATA 07,ee,20,32,6b,1c,02,3a,fe,ae
203 0a DATA 00,b0,00,75,f0,ca,d4,e7,10,2c
204 07 DATA 15,fd,40,6a,9b,b8,ad,93,20,15
205 0g DATA 0c,2a,2d,00,0a,e0,b0,43,fa,2b
206 0b DATA ae,e5,41,b0,7a,db,ec,8b,10,b3
207 0a DATA a9,b7,3a,26,6d,8d,cb,9a,da,36
208 0F DATA 2c,e3,36,b2,d3,42,dc,6c,ae,ae
209 00 DATA 0a,03,32,2e,5b,af,6d,7b,46,2d
210 0E DATA ae,0a,0f,63,32,2e,55,9b,0d,62
211 0N DATA 0a,0e,0a,da,e3,ae,c4,fa,2d,65
212 1a DATA 0a,2f,2f,2f,2f,2f,2f,2f,2f,2f
213 01 DATA e3,ff,43,4c,3f,5e,07,86,97,10
214 0v DATA 3a,47,23,4c,4f,bc,81,04,22,07
215 11 DATA 0a,0b,0c,37,09,cd,22,6b,6e,c5
216 0c DATA 4e,09,ae,28,58,52,2b,0a,66,72
217 01 DATA 4f,00,5b,40,0b,52,41,eb,0d,01
218 0e DATA 0d,ae,5f,4d,00,20,54,4f,0a,4f
219 0a DATA c5,61,ff,21,0d,ff,19,65,45,4a,ff
220 03 DATA 01,0f,53,41,56,21,7e,43,12,33
221 0x DATA 54,92,4f,4e,4a,40,2e,2e,2e,3a
222 18 DATA 7d,ff,8c,39,ff,13,5a,4c,af,43
223 0g DATA 41,42,4e,45,7c,20,03,20,20,60
224 01 DATA 14,13,44,41,54,41,20,2b,20,49
225 09 DATA 4e,46,4f,20,fa,2b,14,90,00,0f
226 03 DATA 41,44,d1,03,53,53,20,52,45,ff
227 3d DATA 52,55,4e,43,48,45,52,20,fe,03
228 01 DATA 78,4f,4e,66,e7,4f,46,46,8d,07
229 4z DATA 2a,21,0c,30,30,80,4f,e0,ff,dd
230 11 DATA 9e,71,ae,ec,9d,ff,ec,e5,19,3a
231 05 DATA 7f,ff,cc,e5,68,01,e2,6f,88,57
232 01 DATA 3d,22,c4,ea,31,36,d8,d5,18,c2
233 0e DATA 20,7a,2a,68,31,6b,1f,44,4a,82
234 00 DATA 77,6e,08,9a,67,8e,e7,03,67,57
235 01 DATA ff,96,c9,1d,b2,ff,22,ff,75,79
236 02 DATA 24,ae,30,30,00,18,8e,0f,0f,65
237 0a DATA c6,85,2a,7c,c0,18,78,3a,37,62
238 0f DATA 52,50,e8,8e,03,66,ca,03,42,64
239 1a DATA 02,42,59,4a,03,66,ff,ff,4e,ff
240 0E DATA bf,dd,08,b5,db,0c,ad,91,fe,f9
241 0a DATA f9,46,ae,05,b6,5a,18,3c,67,f2
242 0D DATA 30,ff,1b,73,50,d8,a3,a3,0a,f4
243 0f DATA 3a,6f,c9,07,6d,e0,ea,95,00,8a
244 01 DATA 0e,27,d0,aa,18,0a,d1,5a,12,16
245 06 DATA 74,fe,03,ae,5d,8e,5a,2b,1a,b8
246 09 DATA 49,b5,fe,6e,03,16,ae,72,69,20
247 01 DATA 0a,0c,40,2e,2f,62,e1,7c,00
248 01 DATA 0d,0e,ae,0a,69,c7,0c,2e,5a,0c
249 02 DATA 84,30,04,1e,80,fe,9d,e7,26,09
250 01 DATA 0b,08,01,72,98,91,ea,09,36,3e
251 0v DATA 35,08,01,fa,6f,41,1a,ac,ff,b0
252 0e DATA 17,88,1c,0a,94,97,fe,61,80,ae
253 0B DATA 13,66,30,3f,c9,bd,fe,4f,90,6b
254 0e DATA 2a,e9,b0,80,65,9a,0a,73,be,9e
255 0a DATA 31,ae,5a,1a,19,bc,28,95,48,02
256 09 DATA 1a,01,fe,27,72,97,0b,dd,30,fc
257 01 DATA 01,7f,0c,06,ec,6e,19,00,32
258 0a DATA 3d,da,ae,21,1a,17,12,0c,4f,11,1e
259 01 DATA e7,31,eb,1a,ae,f0,05,83,13,1c
260 00 DATA 13,49,5f,77,b4,28,15,04,ff,ae
261 01 DATA 3e,c1,0a,b4,51,ba,72,77,63,0a
262 02 DATA 1a,03,03,ae,5b,03,6e,a1,a1,43
263 0e DATA 1a,ad,52,15,01,e7,30,78,7c,ea
264 0e DATA 15,eb,91,ed,af,4d,62,1d,f2,73
265 01 DATA 43,f9,74,92,10,d6,f9,16,8b,99
266 00 DATA 2e,ff,0a,45,9c,15,98,25,fd,65
267 01 DATA 01,1b,f9,12,6a,0c,47,ba,b2,12
268 01 DATA e9,2d,ff,53,2c,53,08,61,7f,6d
269 0a DATA c5,ca,ad,0c,ff,27,7b,de,62,98,fa
270 03 DATA 96,54,9a,9a,fd,3e,19,1f,66,2a
271 01 DATA 34,b0,9b,80,75,4d,96,b6,fb,0a
272 09 DATA 11,9a,4f,be,fd,1b,ec,25,06,fb
273 0E DATA 46,ed,2d,0f,72,26,57,59,fd,05





```

274 le DATA 31,6f,52,f4,9a,07,9b,4c,6d,8e
275 nu DATA ea,e1,1c,5a,30,e7,0b,38,40,03
276 v9 DATA ea,09,0e,6c,8f,84,fc,d1,dc,95
277 nb DATA 0c,d4,4c,0f,0c,0e,0c,0e,0f,0c
278 nz DATA fe,1a,0f,0f,0c,0c,0c,0c,0c,0c
279 lv DATA a4,5a,de,6a,03,f9,0a,0c,3a,fd
280 4p DATA fa,b7,af,0c,0c,0c,0c,0c,0c,0c
281 wN DATA fe,0b,b9,a2,60,2c,6b,4f,81,d5
282 cW DATA 0b,67,43,0b,7e,07,fb,73,2d,51
283 FM DATA 0f,6f,fd,6f,1c,0c,0c,0c,0c,0c
284 1L DATA 63,09,f1,bd,0d,e2,48,56,45,bc
285 w5 DATA fd,8e,d0,fc,70,b2,03,16,79,54
286 J2 DATA ea,15,a1,0b,0f,0c,0c,0c,0c,0c
287 J2 DATA 70,37,0c,0c,0c,0c,0c,0c,0c,0c
288 J3 DATA df,f1,9c,42,f1,78,0c,fb,11,ed
289 14 DATA 11,1f,3c,76,54,21,f8,74,33,95
290 de DATA 0c,fc,1e,0c,0c,0c,0c,0c,0c,0c
291 w0 DATA fe,60,5a,4e,0c,0c,0c,0c,0c,0c
292 JM DATA be,fb,02,d8,0e,6c,4c,e8,af,51
293 4q DATA 1a,9e,0f,56,fb,0c,0c,0c,0c,0c
294 Th DATA 3a,fa,b3,0c,32,3a,1c,1c,fb,da
295 qg DATA d9,18,fd,98,ce,45,ed,c5,fc,20
296 JG DATA 72,51,a1,b1,fe,0e,5c,9a,db,90
297 75 DATA 70,09,45,09,2a,fc,48,60,a7,6e
298 Th DATA 91,f9,2d,4b,0a,7e,2a,e2,07,72
299 Y7 DATA 52,fd,e4,e6,3d,8a,e9,14,ff,57
300 xP DATA 7e,00,02,39,3e,eb,16,4d,9c,59
301 2o DATA d9,7c,60,d1,58,09,18,58,5f,0a
302 QD DATA 0c,31,0b,f4,3e,0c,10,b3,f7,99
303 ud DATA 65,5e,0f,5f,4c,1c,2a,74,1f,77
304 wa DATA 56,60,55,0a,0e,a9,56,aa,55,63
305 w0 DATA bd,65,05,0c,0c,0c,0c,0c,0c,0c
306 4p DATA f8,3b,0c,1a,71,0c,0c,0c,0c,0c
307 wF DATA ed,04,53,4d,1a,0c,0c,0c,0c,0c
308 F2 DATA 0e,6a,2c,07,03,22,6d,51,a9,b2
309 QD DATA 11,5c,5c,d4,21,fa,e4,2c,17,75
310 wN DATA 08,05,38,0b,d5,b0,0e,61,03,94
311 wE DATA 60,d0,3d,08,5e,56,3c,99,ff,48
312 nL DATA b3,f1,4e,fb,0c,0c,0c,0c,0c,0c
313 1L DATA 54,0a,08,7a,6a,0e,61,0b,09,09
314 wN DATA 7c,76,14,9c,83,49,11,f8,9e,00
315 70 DATA 3e,dc,29,bc,4b,16,f8,2a,f3,c5
316 rz DATA 89,e9,f0,6c,9b,0b,db,44,6d,b5
317 Pr DATA bc,ef,34,95,50,d8,fe,32,2a,08
318 CK DATA 17,4f,8d,d1,0c,0c,0c,0c,0c,0c
319 r1 DATA 24,24,9a,0c,0c,0c,0c,0c,0c,0c
320 rT DATA 03,14,95,51,45,ba,ef,12,14,80
321 R2 DATA 30,10,32,11,fc,10,2a,70,aa,7b
322 6r DATA 00,0f,6f,2f,2d,4f,ff,0a,0c,0c
323 x1 DATA 3f,3f,1f,1f,1f,1f,1f,1f,1f,1f
324 eF DATA 32,e1,d0,f3,30,80,32,81,06,eb
325 v8 DATA 51,80,3c,28,20,7a,28,14,20,0f
326 xP DATA 1c,01,2f,0c,0c,0c,0c,0c,0c,0c
327 6r DATA 33,2b,0c,0c,0c,0c,0c,0c,0c,0c
328 gx DATA 5d,16,0b,30,70,02,38,97,0d,08
329 11 DATA 67,06,08,e8,f8,55,03,58,ff,f6
330 Jp DATA f9,16,00,53,2e,2a,6a,ba,80,55
331 bs DATA 09,89,b6,75,2c,24,74,37,77,64
332 c8 DATA 95,b2,c6,3a,c0,6a,89,24,6e,e5
333 J5 DATA 50,10,22,10,d9,06,ef,d9,54,51
334 MS DATA E7,0c,1f,31,73,1c,0c,0c,0c,0c
335 40 DATA 14,4e,1b,80,17,77,2b,21,e8,56

```

```

336 Gg DATA 31,1a,24,38,67,e1,98,85,da,d1
337 vV DATA 2f,9a,ed,cd,db,0b,a2,0d,aa,e5
338 eK DATA f0,b2,59,73,ee,1c,91,f3,dc,00
339 Fx DATA 8a,3c,51,77,00,b6,33,d4,8d,1b
340 zV DATA ab,15,75,b7,61,9b,ba,35,6c,53
341 AS DATA b7,86,6d,ea,6d,b0,4d,dd,09,a0
342 xE DATA f2,04,dd,6a,f2,db,0b,08,38,f3
343 mC DATA db,0b,40,0a,04,b8,97,e2,5e,1a
344 ne DATA b0,41,29,24,56,24,1a,2d,94,66
345 d1 DATA 8b,61,22,bd,84,aa,e3,6b,1a,38
346 xP DATA b4,a9,d9,7a,06,64,73,eb,26,ac
347 1G DATA 2a,b3,d8,00,0c,0c,05,58,d4,77
348 Zv DATA 23,70,8c,17,36,b1,61,36,27,23
349 Qb DATA 70,0b,06,34,74,8a,95,0f,fe,ea
350 J1 DATA 66,26,d0,7e,01,99,8f,ee,7e,02
351 14 DATA f6,e8,7a,50,0c,e1,05,34,4d,d1
352 J2 DATA 8e,83,46,0a,18,72,53,29,b1,26
353 05 DATA 62,6c,26,52,34,8e,9e,9e,b5,62
354 bP DATA 04,d6,a2,ac,16,35,a9,08,8e,cb
355 Th DATA 61,b7,a8,aa,0c,0a,6a,25,f2,4a
356 wV DATA 91,21,f7,22,51,60,05,24,49,43
357 vA DATA e9,79,fe,8a,ad,aa,6a,04,42,19
358 Lg DATA 60,a2,79,6d,01,70,30,ee,3a,04
359 F7 DATA ed,11,d5,a7,26,06,32,a2,48,e7
360 qF DATA 00,22,5e,24,70,40,66,66,8a,01
361 OY DATA 4c,df,44,00,78,5f,04,70,26,a7
362 A6 DATA e6,24,80,42,80,60,f8,f6,45,dd
363 uM DATA 04,4a,99,a2,66,62,00,24,b9,b4
364 BX DATA a8,45,1c,2d,6a,ac,57,b6,43,34
365 pg DATA da,b0,4b,0a,d6,01,59,55,b2,10
366 4E DATA d6,2c,20,49,70,18,23,ef,98,e5
367 Kq DATA c8,21,3d,84,52,25,a0,9c,58,43
368 f2 DATA 23,7a,0a,13,b1,9d,fc,37,4a,53
369 1R DATA 40,60,ee,ce,5b,dd,1a,18,0c,45
370 Gv DATA 3c,7a,16,12,4a,ef,03,06,3b,1b
371 BK DATA a8,81,52,46,8a,07,d7,05,0a,46
372 8r DATA 00,06,65,22,06,5b,59,5b,50,0b
373 wV DATA ed,49,dc,a0,52,0e,5d,5d,65,f3
374 rF DATA 9b,20,51,0e,ff,fe,f9,a9,40,52
375 JQ DATA 3c,07,0b,51,ff,fc,66,ce,83,3a
376 ZG DATA 35,2d,57,92,25,ac,aa,f3,08,00
377 1p DATA eb,14,ff,e6,2a,3f,db,81,5c,f7
378 g8 DATA 28,5e,2e,4a,9e,7a,66,0b,8f,12
379 wF DATA 56,43,13,d2,0b,d5,0c,d5,fb,92
380 Q5 DATA f2,8f,fc,75,85,f2,06,b6,d1,1a
381 bI DATA 03,b0,b9,15,06,58,6e,3a,b6,16
382 1w DATA 21,78,2c,2d,22,32,14,54,21,15
383 nB DATA 26,33,11,0c,17,0a,8d,81,0b,17
384 9s DATA aa,20,ba,f8,14,d8,9e,7e,14,f0
385 vN DATA 6c,14,06,2d,9e,4a,ea,10,9c,a5
386 Tu DATA 82,fb,2e,73,95,98,b6,8e,ab,ed
387 wM DATA 92,74,49,10,5e,ef,02,ad,7f,b2
388 Jc DATA a2,15,6e,01,0e,0c,0a,01,48,18
389 xG DATA 21,7c,67,84,18,45,fa,f6,6c,f7
390 x1 DATA a5,0c,50,87,3a,cd,31,7a,f6,5e
391 1p DATA f5,21,24,08,76,08,39,93,45,36
392 8P DATA 75,84,de,12,20,e2,88,41,e0,15
393 eS DATA 2a,4c,15,22,20,fe,90,b4,2b,82
394 wM DATA fb,1e,d9,15,03,67,22,5d,ed,c5
395 E8 DATA 1e,ee,6b,1d,2a,dc,d0,f8,5b,07
396 MZ DATA 60,d9,70,f4,90,5b,20,98,0c,02
397 su DATA 5a,2d,96,d5,99,aa,14,e4,58,5c
398 vJ DATA 14,da,0a,d0,b3,0c,6e,7a,e2,d8
399 dy DATA fd,ec,ea,3a,ao,27,20,62,5a,56
400 RO DATA 83,43,82,02,d6,22,7a,20,f3,e3
401 wJ DATA b8,08,61,02,aa,66,72,98,49,63
402 xJ DATA 4e,30,6b,32,6a,ba,ae,25,1d,58
403 rT DATA 63,77,0e,b2,19,02,9e,50,c2,fa
404 6r DATA 8c,0c,63,b0,e2,fe,cd,4a,da,23
405 eJ DATA 88,83,fb,59,0a,8b,61,a0,7b,66
406 Pw DATA b8,2b,0b,44,01,da,31,03,fa,06
407 11 DATA 77,d8,fe,55,c0,02,72,3c,45,d1
408 11 DATA 54,ee,42,81,e1,8c,b1,19,6d,48
409 vJ DATA 1f,da,0a,b6,af,6a,4d,db,28,0d
410 aw DATA d2,b5,d9,d0,52,06,80,8c,99,6b
411 J5 DATA 31,0a,8c,48,84,a6,ao,1f,1f,a8
412 v6 DATA 66,14,14,1f,73,8b,1f,9e,96,6d
413 r8 DATA a7,ab,04,21,bc,22,08,59,c7,91
414 TX DATA 03,ed,99,b7,e2,da,21,eb,51,96
415 wV DATA 22,e8,05,4f,d8,22,00,24,db,05
416 vJ DATA 6e,26,3a,1f,72,f8,ef,d6,6a,6a
417 vA DATA 50,01,99,13,c0,99,66,22,c6,fc

```

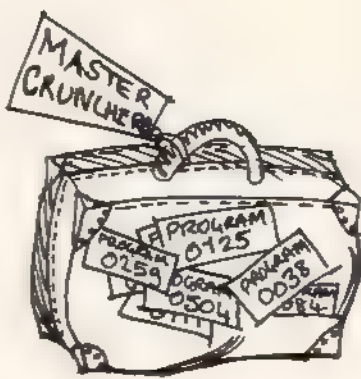
```

418 aF DATA 4a,4c,43,0f,50,a8,8f,50,41,43,4b
419 GJ DATA 37,a1,56,0e,2e,14,4e,f9,51,6a
420 eR DATA 79,98,4a,a8,01,6c,51,7e,e5,96
421 oK DATA 1a,13,ee,aa,b0,0c,15,1f,2d,a1
422 wV DATA aa,03,0c,0c,0c,0c,0c,0c,0c,0c
423 PI DATA 4c,02,0c,51,f6,27,0e,2c,4c,5b
424 Y1 DATA 59,f6,a5,3c,1c,56,02,79,d4,54
425 cK DATA 2a,02,9b,0a,1f,0a,e9,3a,dc,bf
426 tx DATA fa,2f,0b,2b,5a,07,1b,40,06,45
427 J7 DATA ed,f5,1b,7c,f2,0c,5c,47,69,92
428 OW DATA 0c,c4,9c,3a,0c,a8,56,31,2e,32
429 43 DATA 4c,51,4c,6e,ee,0c,1c,02,f2,44
430 1F DATA 53,01,1e,ba,c6,19,71,2b,b9,1c
431 17 DATA 04,ef,af,f6,f6,0c,ae,19,f9,34
432 NM DATA 1a,fa,1c,aa,2c,66,7a,1e,92,01
433 mA DATA 1e,96,35,00,40,66,d9,e3,b3,1c
434 sY DATA c9,2c,0d,10,c0,08,0e,66,5c,ec
435 F0 DATA 61,5e,b5,4b,6c,a8,21,f6,01,23
436 KE DATA 4c,0c,0c,0c,0c,0c,0c,0c,0c,0c
437 c1 DATA 0c,ef,0c,1c,2c,f8,1a,9a,2e,55
438 qK DATA 2c,9a,59,e1,ab,fb,16,0e,22,c3
439 24 DATA b7,1f,0e,2a,1a,2e,42,b3,70,12
440 CW DATA 5c,f7,a0,0c,0c,99,f1,0c,0e,e3
441 73 DATA a8,2c,16,1b,96,9c,1b,aa,60,b8
442 1 DATA dc,01,b5,05,1a,e2,f2,4c,c6,ac
443 11 DATA db,b2,66,4e,79,15,1e,16,87,27
444 9 DATA 4c,1a,fc,0c,33,b8,13,63,7c,7a
445 11 DATA 41,a7,3a,3a,f3,62,83,48,30
446 eW DATA 33,e6,af,11,30,f6,48,0d,d2,80
447 J3 DATA 1a,0c,5b,2b,8c,0c,53,31,0b,71
448 70 DATA 21,11,16,72,27,86,b4,5f,29,ad
449 70 DATA ac,44,13,0c,0c,06,01,f2,2d,bf
450 Jq DATA 0c,61,1d,c2,e6,31,b2,58,8f,b6
451 wV DATA 11,21,1f,07,2c,e7,0c,d1,6d
452 hw DATA 3c,15,d7,1b,0b,4e,67,c6,f2,f1
453 uC DATA 17,33,ed,ed,38,c1,20,7a,1d,78
454 U1 DATA 7c,98,7c,bf,ea,f3,6b,6c,db,43
455 U2 DATA 7c,2f,4a,7e,66,56,22,18,52,81
456 Te DATA b2,0c,66,4e,11,7f,1b,ec,e9,a8
457 BQ DATA c5,ca,f6,ea,a8,c5,c0,f6,eb,a8
458 JB DATA c5,c2,f6,ec,e8,5c,b6,f6,f2,a8
459 F5 DATA 9c,31,00,0a,73,f1,51,8b,14,5a
460 Z1 DATA ef,4a,e2,4c,0c,66,03,70,fc,17
461 CF DATA 4c,70,2c,07,0c,a2,a0,03,c1,0c
462 pB DATA 1d,64,c9,80,d0,fa,ed,10,a0,ea
463 wV DATA 3b,61,81,78,2a,48,22,48,67,63
464 yY DATA 0e,02,80,74,df,00,1d,0e,4c,60
465 O7 DATA ee,2c,af,11,bd,90,8c,2a,48,95
466 JT DATA c9,2c,2a,1a,16,86,73,5b,1c,a2
467 ZF DATA 92,8a,23,c4,ef,f0,32,82,b2,9a
468 Jx DATA c3,fd,9c,20,ad,7b,30,52,96,6c
469 70 DATA da,08,24,50,88,9a,b1,67,06,20
470 5g DATA 16,bc,06,16,ac,5e,88,d1,c0,60
471 OR DATA f2,88,01,11,7e,db,6a,6d,b1
472 KO DATA 04,6e,00,77,f1,22,c1,ae,b4,da
473 8F DATA 17,78,22,2c,7a,fe,0c,07,ac,15
474 2C DATA 28,67,55,32,20,80,fb,9a,aa,57
475 xH DATA 25,2f,et,33,0c,99,33,9a,1a,25
476 Uv DATA 1e,0b,27,0c,2c,7a,0c,2b,4d,a5
477 DV DATA 1e,af,0c,bc,4f,4e,f6,2d,5a,07
478 ze DATA 0c,0a,1c,62,ee,69,0c,1f,6c,42
479 BR DATA 90,1e,29,bc,ab,ee,66,01,93,ee
480 nJ DATA 0b,fe,ef,52,63,05,15,8b,db,0a
481 11 DATA 66,70,09,f5,a3,98,6d,20,f6,d9
482 eG DATA b2,62,01,16,54,db,0c,8e,1e,0c
483 KI DATA 52,40,45,75,5a,7c,f9,6a,eb,4a
484 yD DATA 1b,f8,97,8b,36,5a,02,9a,ec,ca
485 4q DATA 2e,0c,07,07,20,2a,1f,0a,12,fa
486 11 DATA 1e,ca,02,71,0c,31,5f,70,30,7e
487 IV DATA 0b,f9,0c,0c,85,f3,0e,72,fa,05
488 HC DATA 4a,fd,64,4b,03,03,7b,34,f4,6b
489 9e DATA 0a,0c,17,12,6c,07,5a,07,ac,fa
490 On DATA 14,4a,a9,24,0e,65,4c,2a,8b,05
491 7N DATA 6e,02,5e,e0,45,95,6d,03,e2,72

```

Listing. "MasterCruncher... Gen"
(Fortsetzung)

492 Ap DATA 0a,51,83,0b,b9,c1,0e,dc,60,0d
 493 Mx DATA 6e,30,0e,17,4d,01,52,0c,44,fe
 494 HII DATA 47,2b,b0,ff,88,7c,7e,30,94,37
 495 q2 DATA 6c,60,06,79,74,42,50,68,2b,1e
 496 lw DATA 6e,4e,50,66,2a,f4,62,1d,32,b3
 497 de DATA 1b,36,96,a1,b8,04,d4,58,71,60
 498 6g DATA 06,5e,c0,4b,fa,ec,e4,51,de,ed
 499 Yz DATA ec,80,67,66,b1,00,09,40,a9,37
 500 YR DATA 6c,a7,e0,00,20,3d,22,7a,1a,24
 501 JQ DATA 72,00,32,00,ed,48,e1,49,d2,4b
 502 Om DATA 41,f0,68,00,ff,05,43,f1,10,1b
 503 qS DATA 72,07,10,1a,12,80,46,00,10,80
 504 xI DATA d0,00,f8,1f,d2,54,28,1d,b3,68
 505 LG DATA ee,4c,9f,5b,30,61,3e,66,ae,80
 506 xn DATA 44,b5,28,e6,9c,6c,0c,07,1b,ce
 507 zd DATA ff,5a,67,f6,1d,0e,61,a7,1d,be
 508 7a DATA b1,f7,de,0e,4f,d0,61,02,60,d8
 509 aG DATA 91,9f,96,ea,32,80,31,57,ea,60
 510 e0 DATA 14,ce,1a,80,25,0e,44,6a,46,50
 511 CH DATA 83,30,d8,3a,49,e3,b4,ab,08,a6
 512 Bq DATA 5a,0e,8d,f5,4d,08,6b,2d,00,dc
 513 RI DATA 99,3a,61,c0,66,2e,98,6a,01,44
 514 aA DATA 2d,90,6a,01,5d,c0,f6,88,72,01
 515 ZX DATA 34,20,80,6a,01,2d,c1,f6,76,42
 516 YI DATA 80,61,24,c6,45,3a,70,01,61,1c
 517 M8 DATA f9,a2,fc,04,6a,14,16,4a,70,03
 518 fL DATA 61,0c,e5,8b,42,3a,c3,8b,3e,67
 519 p4 DATA 64,3a,1d,24,d2,f2,6a,1c,d6,4a
 520 PM DATA 41,67,06,20,50,53,41,60,f6,4e
 521 50 DATA e8,5f,ff,80,b4,14,01,67,22,4a
 522 69 DATA 18,66,20,e7,7b,1b,c8,70,81,01
 523 Ke DATA 32,de,a2,10,a2,1b,74,30,08,08
 524 wh DATA 0a,6b,d7,19,c2,4a,29,66,fc,53
 525 UJ DATA 49,0c,29,ac,1f,6a,ac,6b,12,56
 526 J8 DATA 43,2f,f2,67,04,12,d8,60,98,42
 527 G8 DATA 19,0c,bf,13,27,01,27,de,95,19
 528 B7 DATA c3,7f,07,58,f8,68,00,14,79,88
 529 Un DATA 14,d6,02,ae,24,b2,8b,9e,6c,be
 530 kF DATA a2,3e,30,1c,a0,73,b0,96,0d,88
 531 Pd DATA b6,04,36,86,42,90,2f,0e,c8,23
 532 pU DATA 1c,3a,22,52,70,3f,42,1a,09,7f
 533 Hq DATA 0c,fa,b3,39,da,67,16,2a,51,70
 534 9v DATA 40,bd,24,af,22,4d,b6,f7,ab,4e
 535 01 DATA 66,ea,2e,5f,8e,42,0d,fd,66,c7
 536 SN DATA 1a,4c,4a,90,ae,7a,30,07,3b,f3
 537 78 DATA 40,53,50,de,83,25,80,6b,1c,36
 538 Dv DATA 2c,f6,4e,a0,34,3b,30,3c,ce,02
 539 dQ DATA 72,c8,fe,ab,a8,54,08,e0,f8,ed
 540 de DATA 10,c0,db,89,00,e8,65,07,22,0d
 541 Nx DATA c6,a0,01,2e,73,07,7e,ae,00,03
 542 9K DATA 00,ac,01,03,f7,76,fe,f4,96,53
 543 oY DATA e4,0f,36,b8,39,43,23,6e,0b,62
 544 T0 DATA ed,a8,a0,2d,e2,74,45,24,d6,ee
 545 87 DATA 42,09,1a,3e,78,45,c0,9b,22,ec
 546 6N DATA 77,ae,61,0e,68,23,44,48,37,35
 547 h1 DATA ba,fe,0a,ec,fd,b2,10,ae,8e,de
 548 TT DATA 12,55,c1,3e,2d,7d,10,c0,10,c1
 549 T1 DATA 3e,f1,0e,24,1a,e9,12,f0,1c,58
 550 Kp DATA ac,29,92,f4,12,25,6e,b7,60,9e
 551 W2 DATA 2d,8d,44,24,8b,47,fa,02,70,36
 552 ge DATA bc,cd,ff,46,11,9c,16,32,4a,10
 553 J5 DATA 66,0e,e5,1d,18,28,11,7c,00,3a
 554 t4 DATA 7a,bf,e9,19,c3,61,94,14,6a,99
 555 TQ DATA b5,62,1d,c1,8c,14,5a,b3,8f,06
 556 TG DATA 46,98,0b,c6,99,e9,30,3c,24,98
 557 PC DATA 82,61,f7,f1,fb,d4,96,b3,c2,0c
 558 KX DATA 90,76,39,05,65,26,20,3a,1a,b6
 559 V1 DATA 5b,80,b0,f5,4b,ac,67,f0,92,05
 560 SF DATA ba,2c,0f,a2,52,90,6d,3a,11,20
 561 oJ DATA ba,1a,98,70,f4,01,af,63,fe,66
 562 1q DATA b9,bc,0e,e0,04,15,c6,88,66,42
 563 2b DATA 82,42,83,76,23,30,59,40,3b,65
 564 3L DATA e6,1a,d0,4c,98,00,d6,2f,17,36
 565 DM DATA 64,e2,e8,38,17,30,65,d2,8e,17
 566 Tc DATA 2c,64,d6,e3,38,17,26,65,d0,4a
 567 vq DATA bc,d3,c0,da,6e,60,30,61,68,36
 568 xn DATA 1f,dc,d8,a5,a0,66,3e,f3,fc,64
 569 Mg DATA f0,d5,be,f6,65,ea,da,57,f2,64
 570 Z2 DATA e4,fb,ec,65,de,61,3e,2a,af,16
 571 nI DATA e2,64,ca,b4,2d,9e,d2,65,e4,b2
 572 JU DATA c0,09,d8,64,b6,3c,d2,65,b8
 573 2Q DATA a5,ab,ea,de,60,02,bd,21,45,24



574 Rq DATA 10,4a,52,67,02,61,0a,62,9f,8e
 575 99 DATA 06,42,cb,2c,92,c9,35,2c,7a,16
 576 58 DATA 94,66,27,15,d4,e1,26,f0,58,42
 577 rJ DATA 28,02,2a,03,54,40,d9,fe,eb,24
 578 fv DATA 00,26,01,96,80,9a,81,55,44,2c
 579 -r DATA da,2f,82,50,b3,05,b8,a2,32,08
 580 -u DATA 63,65,01,24,66,8c,d3,95,43,52
 581 6F DATA 66,30,29,1d,1a,9a,56,82,55,64
 582 qV DATA 25,e5,00,e3,99,93,11,9a,54,30
 583 Ab DATA 01,12,2e,2b,08,20,49,0a,79,95
 584 -e DATA 02,17,0a,5b,00,05,9d,d2,00,5e
 585 zL DATA 00,e1,94,8f,06,a1,4e,35,00,76
 586 Xb DATA 01,3d,00,83,f7,01,77,3b,37,70
 587 fP DATA 9f,43,57,6b,39,46,b6,49,19,38
 588 -e DATA e9,75,14,49,28,30,0b,f0,d8,58
 589 -u DATA 26,65,00,91,8e,ae,18,05,06,0d
 590 -e DATA 5e,68,db,68,73,00,0e,de,9a,f0
 591 e2 DATA f6,b5,89,86,00,7a,18,84,67,fa
 592 01 DATA 3e,07,af,3b,f1,00,45,50,51,52
 593 -s1 DATA 53,54,55,56,57,58,59,00,46,5f
 594 -v2 DATA 4c,4d,4e,4f,42,00,30,31,32,33
 595 -v4 DATA 34,35,36,37,38,39,61,41,62,42
 596 96 DATA 63,43,64,44,65,45,66,46,e5,0d
 597 -t DATA 7f,01,ce,11,39,8a,11,36,44,4b
 598 21 DATA e8,f2,aa,9c,40,53,21,42,84,38
 599 -u DATA 6b,18,2c,00,53,00,51,67,fc,42
 600 -g DATA f3,ed,cf,e4,e1,3f,ac,3c,64,18
 601 -X DATA 4a,05,76,f8,30,81,9c,32,3a,18
 602 -L DATA 7c,7d,13,fe,38,6c,0c,af,c1,ce
 603 -v DATA 8a,32,2f,ce,60,43,f1,3b,f2,9e
 604 -3 DATA -1,66,28,4a,44,67,c8,53,44,91
 605 SC DATA 7f,65,ba,53,13,8d,7e,53,45,96
 606 -u DATA -2,12,2b,17,57,73,4c,eb,3a,03
 607 -x DATA 52,60,78,dc,5c,8f,fa,4d,6d,76
 608 -fr DATA 2c,6e,1e,1e,1e,17,aa,0a,17
 609 -q DATA b4,72,20,75,ae,b8,3e,e5,66,0b
 610 -7 DATA b1,17,22,17,84,0c,13,76,ae,9e
 611 -J DATA ac,45,14,4a,53,67,18,99,b8,2a
 612 -6A DATA -4,4f,11,3a,30,5c,72,15,f0,33
 613 -3 DATA -0,5e,7e,21,7a,14,f1,ac,d3,b8
 614 -2F DATA 06,78,59,ef,fa,f1,65,04,74,d4,fd
 615 -v DATA 0a,10,ba,17,6e,51,4f,fd,06,52
 616 -6 DATA 93,e1,03,c4,01,52,44,d1,05,d8
 617 -V DATA b4,2f,de,47,ee,21,52,1d,cd
 618 -PC DATA c2,ae,96,3c,3e,21,4c,70,72,3f
 619 -v DATA 74,b7,7c,db,99,ba,01,1a,ec,20
 620 -X DATA 2d,1d,fe,5c,9a,02,b2,2a,7a,fc
 621 -Mc DATA e5,31,15,26,9b,40,17,e2,4a,50
 622 -Rz DATA 67,72,0c,50,e0,2f,9f,60,b5,86
 623 -M DATA c0,46,d1,35,aa,a6,00,6a,ac,04
 624 -zs DATA 4a,4a,40,67,40,d0,17,94,00,02
 625 -8S DATA 80,20,7a,b6,00,14,02,24,88,11
 626 -fD DATA 17,c2,d1,92,99,77,d6,a8,f5,c2
 627 -Vp DATA 01,b3,e5,c8,8e,0d,ef,04,b8,ab
 628 -fD DATA 49,02,59,ef,fa,f1,65,05,44,45
 629 -4p DATA fa,ed,c0,34,bc,ee,6f,8c,ca,78
 630 -2N DATA 9a,d4,00,10,d2,39,38,a9,49,18
 631 -Ez DATA -1,2d,25,0d,2c,6c,00,ee,0c,79
 632 -8Q DATA 01,e9,05,38,17,96,d0,00,11,b9
 633 -vM DATA 03,4c,da,e7,ef,7a,02,06,f0,76
 634 -yC DATA 80,dd,bf,80,72,06,c8,12,25,c0
 635 -nE DATA 01,24,fc,94,d7,98,de,48,0d,ed
 636 -Th DATA 51,b5,20,3a,13,4e,f0,41,1f,10

637 14 DATA fe,00,13,79,eb,98,00,fa,19,5e
 638 01 DATA e8,9f,5c,6b,b6,2b,13,9d,b5,44
 639 7W DATA d1,6a,8f,08,d5,03,0d,00,66,5c
 640 VM DATA 80,07,1d,c6,a1,10,28,10,1b,46
 641 wp DATA 00,e2,08,65,26,f4,01,40,4a,00
 642 zw DATA 4d,01,ff,9b,ea,32,80,60,2e,94
 643 -vZ DATA df,e6,e0,01,e9,0d,60,20,c2,43
 644 -eZ DATA 67,62,1a,0c,9d,59,6e,08,96,93
 645 JC DATA c2,51,60,1c,a7,40,00,70,c1,15
 646 -xx DATA be,01,a9,ad,97,ec,6a,bf,0e,19
 647 -ce DATA e7,df,39,1b,7c,c0,31,b5,42,81
 648 -x5 DATA 7a,04,dc,68,12,18,42,82,2e,2a
 649 1 DATA 05,5a,a9,ba,80,82,58,45,f8,a4
 650 4m DATA ec,1a,e8,98,71,de,82,fc,d5,45
 651 bS DATA 03,3e,ec,41,00,12,14,29,16,19
 652 JU DATA 73,1f,01,04,b2,03,66,f2,37,1f
 653 -Ll DATA 4f,30,01,31,02,32,03,33,04,3a
 654 -Ll DATA 05,35,06,36,07,37,08,38,09,3b
 655 -Pg DATA c0,7f,49,30,0a,41,0a,41,0a,41,75
 656 -dA DATA 0b,62,0c,43,0c,63,0d,44,0d,64
 657 -ad DATA 0e,45,0e,65,8f,46,0f,66,1a,e0
 658 -oT DATA bf,65,ed,f0,80,74,00,e9,98,16
 659 -oM DATA 7e,f9,10,0f,74,ff,0e,38,09,62
 660 -Fg DATA 06,41,0f,30,60,04,06,03,00,37
 661 -eZ DATA 10,c3,cd,7f,7d,e2,ab,8c,01,0f
 662 -kz DATA 18,0e,ae,1a,ae,12,0e,b4,e8,ff
 663 -xP DATA c2,58,b6,12,0a,93,68,02,02,25
 664 -mX DATA 10,12,04,2a,80,b8,cf,76,62,4a
 665 -vD DATA 55,66,3a,22,7a,11,de,20,09,f8
 666 -pC DATA 77,cc,30,6a,35,45,47,fd,96,a0
 667 -vE DATA 43,5e,61,19,5a,08,81,61,28,7a
 668 -vM DATA e4,61,22,68,0f,c7,61,1c,ed,a1
 669 -dL DATA ea,61,16,3d,3b,d4,a2,d0,8a,10
 670 -9J DATA 4a,b9,98,2a,e5,83,7e,0b,42,80
 671 -JR DATA 10,19,0e,98,e3,f7,9e,31,54,56
 672 IM DATA 48,51,0f,ff,0f,58,5f,0a,41,97
 673 OR DATA 9c,00,df,f0,06,66,f6,f9,3e,52
 674 -vu DATA 67,fe,b7,cd,03,cc,af,1e,43,e8
 675 -4a DATA 28,6e,3a,b8,a1,00,67,48,fe
 676 -IS DATA a6,d0,b5,42,c1,03,40,c0,4b,93
 677 -ar DATA ea,eb,52,49,24,5e,b4,83,68,47
 678 -qM DATA 70,e1,8e,65,90,1b,13,6e,a1,70
 679 -34 DATA 00,08,60,4d,30,73,28,48,45
 680 -hX DATA e1,46,60,24,11,b8,34,72,70,e1
 681 -Zt DATA 1c,12,1a,48,80,34,01,e1,49,ed
 682 -gD DATA 4a,d2,42,d2,40,00,f8,0f,10,2e
 683 -pD DATA 3c,56,18,9c,13,b7,14,1c,67,3c
 684 -8W DATA 0c,02,3d,e3,67,da,94,2a,bc,75
 685 -PQ DATA e5,1a,28,34,99,0f,6d,e6,4a,26
 686 -EG DATA 6a,7f,57,47,f3,20,00,42,43,78
 687 -Q1 DATA 07,1a,13,46,05,11,85,fe,4d,06
 688 -fP DATA 43,00,28,d6,ea,d7,77,4b,63,ee
 689 -rP DATA 52,48,6a,2f,e6,c0,5a,61,66,37
 690 -zY DATA 43,74,cb,a3,4e,ab,33,22,5b
 691 -uK DATA 92,54,52,20,3a,10,52,90,81,90
 692 -vM DATA ba,10,48,84,fd,47,2e,72,f9,91
 693 -Mm DATA fe,02,2d,0d,c2,03,11,94,3b,37
 694 -M DATA 0b,43,28,2a,b9,80,10,36,2a,8b
 695 -OY DATA fa,9e,3a,bc,b5,8d,70,65,3b,ad
 696 -1X DATA 31,f6,b8,5b,de,54,44,0c,55,ce
 697 -G8 DATA fe,8b,c0,08,13,7c,42,55,ce,ef
 698 -re DATA 8c,c2,43,f4,0d,f2,b6,c1,06,24
 699 -2K DATA 42,de,36,d8,ae,06,ae,5b,02,08
 700 -ND DATA 0f,ef,42,95,6e,f1,aa,4b,fa,e1
 701 -pB DATA b0,d0,d3,e2,ec,70,00,38,b5,14
 702 -7V DATA c2,b5,be,a8,45,b2,d2,e2,ae,f9
 703 -JM DATA 0f,b6,b8,81,c6,d6,e5,2d,4c,d0
 704 -J2 DATA b6,ba,45,f0,08,00,22,78,e7,de
 705 -5H DATA 50,89,42,45,f1,80,e3,ff,20,08
 706 -90 DATA a3,c6,1f,c7,bf,e0,31,8a,98,6a
 707 -4V DATA 10,10,33,fe,89,bf,84,90,72,01
 708 -1F DATA 24,3a,e6,2c,b0,30,0d,7e,66,36
 709 -oZ DATA 48,e7,80,f0,42,40,43,85,fc,0f
 710 -JF DATA 2e,49,b3,08,66,10,b3,0a,62,0c
 711 -IG DATA b1,eb,62,08,52,40,0c,e0,ff,4d
 712 -3c DATA ff,66,ec,b0,ac,33,82,49,71,33
 713 -49 DATA 59,08,24,ba,33,db,3a,00,24,e2
 714 -1B DATA 5e,59,70,52,41,47,f0,10,00,b7
 715 -mD DATA ce,57,ca,ff,ba,e2,3f,62,32,41
 716 -Lr DATA 86,5c,90,41,e9,a1,01,67,32,bd
 717 -gi DATA 52,0c,b7,41,68,5d,82,0a,10,57
 718 -BY DATA 63,1a,b5,41,56,0d,ea,6a,00,f2


```

719 7k DATA 7a,00,1b,56,e4,41,f0,c6,fc,b0
720 21 DATA 16,60,29,47,ea,ff,ff,b7,c8,b9
721 0j DATA ef,89,4a,12,d8,52,45,e5,af,60
722 gj DATA 2a,0d,e7,fb,9a,41,20,09,e2,88
723 Ns DATA 64,06,52,49,f9,53,65,0f,32,c1
724 Go DATA 20,9d,64,7a,b8,00,f4,93,c8,21
725 mt DATA 49,e4,cb,ce,ee,7c,ac,22,b3,2a
726 H3 DATA 76,05,e1,96,a6,ba,71,39,f6,9a
727 36 DATA 30,77,d0,02,06,48,82,6d,03,e5
728 vQ DATA a9,82,45,42,88,af,30,07,23,77
729 BT DATA a2,50,e5,05,08,0f,da,a6,93,23
730 1e DATA 02,05,03,08,0a,12,dc,6f,cf,2c
731 A7 DATA 76,02,b6,38,83,db,b3,e2,4b,5c
732 at DATA 44,55,b4,ac,a3,4b,36,33,30,06
733 wQ DATA 38,03,e0,4b,e4,5f,44,e7,69,6c
734 R9 DATA 02,44,00,0f,22,2f,a0,07,16,b2
735 6x DATA 01,20,06,06,09,0a,0c,0d,f8,f1
736 uv DATA 02,3f,62,d5,20,44,32,00,ea,1d
737 90 DATA 3f,6c,0a,04,41,5a,1e,66,78,09
738 eQ DATA 09,e1,3a,3a,78,06,b4,73,67,34
739 mS DATA 2d,04,06,0a,01,01,02,03,0a,47
740 zk DATA fa,ff,f4,e0,3f,d2,38,7a,0a,db
741 p3 DATA 4b,b5,d0,b0,44,5c,ed,ff,f6,90
742 Vq DATA 44,18,33,50,05,48,b4,72,3a,0a
743 HJ DATA 69,d8,a2,f3,44,d2,45,e2,99,ei
744 pL DATA 17,51,ce,00,08,12,7f,42,47,7c
745 2t DATA 07,5e,f0,3f,82,f0,d6,44,1e,39
746 M6 DATA 13,ff,1d,ba,cc,06,1a,70,02,16
747 11 DATA 41,50,37,c9,3a,2a,22,08,24,f2
748 OQ DATA 7d,03,ee,e4,1a,e2,4e,80,67,5b
749 hv DATA f2,c5,d6,b6,9a,aa,c1,5a,b6,d9
750 XN DATA 7a,5a,30,70,14,99,3a,0d,d8,41
751 Ov DATA 84,56,ce,99,72,3f,5a,4e,f6,31
752 GF DATA 29,3a,33,fa,e2,dc,da,be,9e,40
753 tg DATA ce,c9,2a,ba,18,43,51,01,b6,22
754 jo DATA d5,58,de,d8,64,26,92,cb,b6,70
755 V2 DATA 80,7f,b2,43,c2,54,2e,1d,f4,b2
756 9k DATA 8a,23,19,50,66,04,26,8a,23,c7
757 10 DATA 74,21,53,7b,25,26,92,cb,b6,70
758 iV DATA 06,63,03,22,c0,38,03,66,f4,8f
759 JK DATA 13,0e,20,c8,25,b4,88,04,c8,00
760 Qp DATA d0,85,b8,8e,ab,08,16,e9,66,28
761 Qe DATA 28,5d,cb,eb,c0,35,c2,68,bb,1e
762 SD DATA 9a,e0,6e,20,4c,20,39,f9,ae,c2
763 Nu DATA 72,4e,60,60,4a,47,6e,ae,66,2a
764 Bn DATA 60,bb,01,16,20,2d,b8,06,05,23
765 fh DATA c0,cc,73,da,f4,40,40,20,3d,38
766 Nm DATA 8d,20,f7,00,fa,c0,62,22,41,c0
767 gn DATA 30,60,1a,0c,40,3c,93,66,14,f8
768 Wv DATA da,20,1d,56,aa,47,c3,21,ba,0e
769 OM DATA 70,08,cb,4d,14,65,66,9d,b1,82
770 pD DATA 52,57,0c,e8,70,04,4e,2d,03,b0
771 Bt DATA 73,2a,9c,cc,24,08,26,00,22,3a
772 xv DATA 00,c0,b4,fc,f5,d0,f4,5d,6e,e6
773 M6 DATA 33,6b,8a,80,28,b4,ce,2f,b3,6e
774 Vt DATA 67,7d,07,78,11,14,c0,89,a3,21
775 JV DATA c0,26,9b,16,03,eb,b9,2f,20,02
776 eS DATA be,13,03,f3,be,08,3a,0c,2f,98
777 Er DATA 38,7d,02,9a,85,af,a3,14,1a,39
778 1s DATA 80,64,6f,e7,76,1c,80,64,66,31
779 JL DATA 3a,68,73,ef,03,68,26,09,a3,03
780 NP DATA 38,ec,08,03,f0,24,51,0c,09,e3
781 M6 DATA 00,d5,ca,58,8a,2b,4a,7a,0f,70
782 sG DATA 30,61,3b,be,a2,3b,d6,14,40,bd
783 JM DATA 2e,78,31,03,8a,33,36,18,41,a7
784 vS DATA 1b,2e,00,f6,05,ac,06,cd,03,2b
785 Da DATA 2b,0c,27,a1,85,51,fc,e5,59,f3
786 et DATA 89,54,81,22,c1,e0,3f,60,f6,2f
787 Td DATA 0b,2c,07,53,66,26,f1,eb,4b,63
788 mo DATA 31,22,3c,c4,d1,3c,33,58,8b,14
789 gX DATA 53,c9,1e,67,04,08,c1,00,01,fa
790 uS DATA 2a,72,3a,28,c0,58,8c,51,ce,ff
791 xX DATA e0,26,5f,c7,fd,ed,22,e3,e9,a8
792 oX DATA c5,9a,f6,ee,a8,c5,90,f6,eb,a8
793 VT DATA c5,a8,f6,ee,a8,d3,b0,6a,b0,0c
794 xi DATA 6b,d4,a3,bd,02,4e,71,2d,7a,00
795 DQ DATA 23,d6,be,86,6e,c6,2f,07,52,4f
796 bJ DATA ff,18,c3,11,10,5a,2b,73,52,87
797 De DATA 53,87,67,1a,e8,15,a5,04,e8,0e
798 C8 DATA 22,92,4b,28,00,08,58,8a,4e,88
799 08 DATA 23,40,f9,11,48,60,e2,2e,1f,42
800 FI DATA a9,f4,e0,43,d2,57,2a,f6,50,50

```

```

801 LP DATA 68,06,e0,21,34,07,58,0b,41,32
802 iv DATA 4b,51,ac,84,11,34,ff,2e,e1,3e
803 JV DATA f9,2d,1d,61,28,66,07,1b,e5,88
804 7U DATA 20,4b,d7,c0,fa,75,0c,22,64,89
805 10 DATA 22,71,94,ce,89,4e,ae,f3,90,20
806 2e DATA 5f,3e,61,06,b4,81,85,40,ff,36
807 xa DATA 4a,65,67,02,52,86,7a,01,fa,51
808 4a DATA 4a,9b,c4,52,20,06,e7,86,22,70
809 7 DATA 08,f6,21,1b,1b,67,24,b4,e7,89
810 7c DATA 04,bf,c0,28,70,18,00,50,c8,53
811 7a DATA 80,20,4c,24,49,22,1b,10,80,ff
812 7 DATA 46,22,12,d1,c1,24,88,51,c8,ff
813 7 DATA f0,60,d8,e3,1f,06,f0,48,e7,ff
814 7 DATA fe,4b,fa,01,80,41,fa,ff,f2,22
815 ea DATA 50,b0,c0,7f,83,c9,58,89,2b,49
816 7 DATA 80,1e,2b,7c,00,a3,cb,a5,18,61
817 7 DATA 0c,4e,b9,80,f7,81,4e,ff,7f,ff
818 7 DATA 69,2d,20,26,14,2a,18,22,6d,00
819 7 DATA 18,83,c5,d1,d0,91,01,ff,95,b1
820 7 DATA 04,4a,60,6a,02,53,88,90,3f,14
821 7 DATA 51,5a,17,42,91,9d,21,38,76,03
822 7 DATA 6e,15,19,20,3c,82,78,ff,e5,6c
823 7 DATA 46,44,53,42,e9,ff,13,c0,5b,4a
824 7 DATA 43,67,06,b8,41,56,a1,ef,da,14
825 7 DATA 21,96,04,48,82,d2,42,12,df,0a
826 7 DATA 0a,43,2e,02,0e,07,04,01,13,20
827 7 DATA 33,c8,00,df,f1,60,51,ff,57,f6
828 7 DATA da,d6,26,6b,00,14,50,8b,b1,cb
829 7 DATA 6f,54,bf,c8,8e,29,3a,74,03,ec
830 7 DATA 10,01,ca,cb,1a,42,41,52,42,5e
831 7 DATA 00,67,12,b5,29,53,43,f8,33,51
832 7 DATA 2e,03,16,33,20,05,48,83,42,43
833 7 DATA 2b,2f,c1,80,02,e1,4a,2e,6
834 7 DATA 04,03,02,c0,41,f5,69,67,42,47
835 7 DATA d6,67,34,76,01,6a,0c,00,3c,16
836 7 DATA 52,3b,2a,8e,12,c0,7f,00,4a,28
837 7 DATA 6e,0f,00,0f,0f,0f,0f,0f,0f,0f
838 7 DATA 04,83,06,00,32,0b,04,07,88,4f
839 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
840 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
841 7 DATA 66,04,10,20,e3,10,83,52,51,cb
842 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
843 7 DATA c1,d5,c1,53,41,13,22,51,c9,ff
844 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
845 7 DATA 2d,80,5a,a8,f8,06,00,fe,00,fe
846 7 DATA 00,82,e8,87,8c,00,f4,00,86,00
847 7 DATA fe,00,93,00,f4,00,69,80,6e,80
848 7 DATA 04,0f,0f,0f,0f,0f,0f,0f,0f,0f
849 7 DATA 71,0f,0f,0f,0f,0f,0f,0f,0f,0f
850 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
851 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
852 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
853 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
854 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
855 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
856 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
857 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
858 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
859 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
860 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
861 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
862 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
863 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
864 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
865 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
866 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
867 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
868 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
869 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
870 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
871 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
872 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
873 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
874 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
875 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
876 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
877 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
878 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
879 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
880 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
881 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
882 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
883 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
884 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
885 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
886 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
887 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
888 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
889 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
890 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
891 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
892 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
893 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
894 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
895 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
896 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
897 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
898 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
899 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
900 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
901 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
902 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
903 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
904 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
905 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
906 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
907 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
908 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
909 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
910 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
911 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
912 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
913 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
914 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
915 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
916 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
917 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
918 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
919 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
920 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
921 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
922 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
923 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
924 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
925 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
926 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
927 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
928 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
929 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
930 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
931 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
932 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
933 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
934 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
935 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
936 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
937 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
938 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
939 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
940 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
941 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
942 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
943 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
944 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
945 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
946 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
947 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
948 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
949 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
950 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
951 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
952 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
953 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
954 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
955 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f
956 7 DATA 0f,0f,0f,0f,0f,0f,0f,0f,0f,0f

```

```

862 Fu DATA 7b,62,80,39,c6,79,20,56,33,2e
863 ws DATA b8,a3,3c,3c,3d,3d,2d,2d,c0
864 JX DATA 2c,fa,03,13,59,c8,8a,82,3e,50
865 X DATA fa,42,b1,3c,16,52,41,48,4d,eb
866 7e DATA 08,4d,49,e4,41,45,4e,94,66,f0
867 8H DATA 20,f7,52,32,49,05,fe,c0,43,4f
868 8 DATA 5c,53,52,47,47,46,8d,ff,42,c0
869 7E DATA 71,4d,39,cb,22,45,26,e4,f9,45
870 7H DATA 43,48,4e,49,4b,7a,08,4e,4f,8b
871 7v DATA a6,1c,23,19,16,ce,44,31,33,e2
872 7b DATA 05,2a,38,8c,ff,18,0a,60,75,17
873 7t DATA 4f,70,6f,0e,84,53,1f,83,23,e7
874 7b DATA c1,13,1e,aa,00,ef,85,12,12,bf
875 7a DATA 55,80,f7,42,81,13,b7,0d,6e,50
876 7a DATA 73,b1,c1,14,5b,15,ed,55,4d,10
877 wo DATA 0f,99,12,f0,15,56,05,78,20,f6
878 7a DATA 47,44,bf,44,e5,53,53,56,9f
879 7a DATA 16,2c,53,3e,05,2a,5a,49,4f,4e
880 7e DATA 20,88,ff,21,54,48,25,06,b2,35
881 7a DATA 4a,12,af,ff,01,18,9a,0b,e3,20
882 7a DATA 44,55,4d,50,bc,9a,db,c0,62,52
883 7a DATA 43,17,5f,50,5a,f0,a0,20,b4,24
884 7a DATA c0,c1,00,ee,09,72,42,b8,6c,3a
885 7a DATA 53,59,c1,e3,91,c0,3b,c1,c0,6b
886 7a DATA 90,83,b6,c0,24,bb,60,1f,0e,67
887 7a DATA 2f,78,67,40,1e,58,20,c1,42,08
888 7a DATA 54,22,20,02,9a,05,09,49,53,20
889 7a DATA 41,4c,9a,7f,41,44,59,52,7f,5b
890 7a DATA 13,1d,77,80,c7,02,00,0e,90,02
891 7a DATA c1,ed,0c,1f,c9,00,1f,c0,46,2e
892 7a DATA 96,00,4a,1d,02,5e,01,d6,32,4e
893 7a DATA 13,06,72,3d,2c,4e,4f,b8,65,53
894 7a DATA 41,c4,68,44,92,89,00,af,0c,d3
895 7a DATA 43,52,61,43,48,16,fe,47,91,4b
896 7a DATA 50,96,41,53,84,43,57,41,49,54
897 7a DATA 7c,aa,00,61,44,02,41,42,ef,7d
898 7a DATA 20,54,4f,20,52,45,50,41,43,f9
899 7a DATA 1d,64,40,4c,45,ac,cd,71,07,4f
900 7a DATA 55,54,f8,96,20,4a,43,4d,e1
901 7a DATA 2f,59,a4,32,60,03,20,20,ca,27
902 7a DATA 01,0e,20,0d,52,e1,2e,2e,2e,20
903 7a DATA 04,27,44,45,4b,4b,4b,4b,4b,4b
904 7a DATA 55,4e,4b,3d,52,20,4f,56,45,52
905 7a DATA c0,41,59,45,44,20,00,1f,14,00
906 7a DATA 50,ff,43,ff,1f,18,bd,81,0a,c0
907 7a DATA 2e,67,4d,70,68,69,63,73,2e,6c
908 7a DATA 49,62,72,61,72,79,e8,7f,1a,37
909 7a DATA 03,0e,39,1b,30,43,f8,1e,72,f0
910 7a DATA 3c,31,36,3c,e4,40,1d,32,3a,bb
911 7a DATA a6,c1,ff,01,1e,13,3b,b5,04,0f
912 7a DATA 78,0c,1d,50,40,20,30,30,00,74
913 7a DATA 6f,70,61,7a,2e,66,6f,6a,74,f8
914 7a DATA 13,1d,6c,0c,09,93,66,05,7a,c6
915 7a DATA 37,44,46,30,3a,74,20,32,0f,80
916 7a DATA 2a,ef,58,05,0f,af,4b,40,9e,f3
917 7a DATA f3,63,6f,6f,73,6f,6f,6f,6f,6f
918 7a DATA 69,76,69,63,65,f7,1f,c4,e1,19
919 7a DATA c0,32,ee,ff,80,18,3c,e0,01,0f
920 7a DATA 04,0c,3b,8b,8b,00,49,ae,50,55
921 7a DATA 76,64,ec,3a,1f,a8,5a,15,92,70
922 7a DATA 7e,3e,6a,0a,5c,15,0c,16,50
923 7a DATA 20,07,c6,35,ce,a8,02,22,d6,10
924 7a DATA 06,d4,a0,3a,0a,23,44,70,d0,50
925 7a DATA 83,e6,1a,f0,d4,a0,f6,0a,24,fe
926 7a DATA 70,08,50,83,0e,1a,16,d4,00,1e
927 7a DATA 06,30,35,48,a8,41,52,0d,58,6a
928 7a DATA e6,50,43,9e,a8,ac,4a,25,0c,01
929 7a DATA 0e,36,aa,2c,62,0c,0b,11,e2,30
930 7a DATA 50,46,29,28,eb,aa,00,01,00,00
931 7a DATA 03,f2,40,f8,00,00,00,00,00,00
932 7a DATA 03,eb,70,70,7d,f1,c0,0c,03,f2
933 7a DATA 00,00,03,e8,00,00,00,03,50,49
934 7a DATA 55,53,00,00,24,7a,c7,a3,fb,0c
935 7a DATA 20,0c,03,ff

```

Listing: »MasterCruncher_Gen«
(Schluß)

Es existiert bereits eine Vielzahl von Anti-Virus-Programmen, doch jedes hat seine Schwächen. Eines findet nur bestimmte Viren, ein anderes hilft nicht gegen Link-Viren (dazu später). Mit »VirusControl« bieten wir Ihnen ein optimales Werkzeug gegen Viren jeder Art an. VirusControl ist trotz seiner Leistungsvielfalt einfach zu bedienen, denn gerade für Einsteiger sind Viren auf dem Amiga gefährlich. Erstens weiß er die Anzeichen nicht zu deuten, und zweitens ist er meist nicht in der Lage, sie zu beseitigen.

Damit Sie die einzelnen Erklärungen richtig verfolgen können, sollten Sie zuerst das Programm besitzen. Folgende Schritte sind erforderlich, um VirusControl zu installieren:

- Stellen Sie sicher, daß auf der Zieldiskette mindestens 35 Blocks frei sind. Die Anzahl der freien Blocks erfahren Sie, wenn Sie in einem CLI-Fenster den Befehl INFO eingeben. Wenn nötig, müssen Sie unbenötigte Dateien von der Diskette löschen.
- Starten Sie den Amiga-Basic-Interpreter durch Doppelklick. Geben Sie folgende Kommandos ein:

```
CLEAR ,50000
```

```
LOAD "VirusControl_Gen"
```

- Starten Sie das Programm mit dem Kommando:

```
RUN
```



DER ENTSEUCHUNG

- Wenn das Programm den Text »Bitte Workbench-Diskette in DF0: einlegen!« ausgibt, legen Sie Ihre Workbench-Diskette ein, mit der Sie normalerweise den Amiga starten.

Auf dieser Diskette befindet sich nun im Verzeichnis C das fertige Programm. Außerdem ist die »Startup-Sequence« verändert worden, so daß VirusControl automatisch gestartet wird.

Um sicherzugehen, daß bei Ihnen nicht schon ein Virus im System ist, schalten Sie Ihren Amiga nach Beendigung des Programms für 20 Sekunden aus. Booten Sie anschließend mit der behandelten Workbench-Diskette. Sollte nun etwas Außergewöhnliches passieren, lesen Sie weiter, denn es folgt die Erläuterung der Funktionen von VirusControl.

VirusControl wird im CLI oder in der »Startup-Sequence« gestartet. Auch ein Start von der Workbench ist möglich. Die Parameter werden entweder beim Start im CLI angegeben oder in die »Info«-Datei im Feld »TOOL TYPES« eingetragen.

Beim Versuch, VirusControl zu starten, kontrolliert das Programm, ob der Amiga bereits infiziert ist. Hierzu werden einige System-Vektoren überprüft. Wenn diese verändert sind, wird VirusControl nicht installiert. Statt dessen erscheint die Aufforderung, den Amiga auszuschalten, wieder einzuschalten und möglichst frühzeitig VirusControl zu starten. Nun kann kein Virus mehr unerkannt in das System eindringen. Bei Veränderungen an einigen bestimmten System-Vektoren gibt VirusControl nur eine Warnmeldung aus, da diese auch von »seriösen« Programmen (Facc II, XOPer etc.) verbogen werden. Ferner prüft VirusControl, ob es bereits installiert wurde, und gibt eine entsprechende Meldung aus.

Bei einem Reset wird geprüft, ob die linke Maustaste gedrückt ist. Ist dies der Fall, wird VirusControl entfernt. Ansonsten zeigt ein weiß-blaues Farbsignal an, daß VirusControl weiterhin das System kontrolliert. Ab jetzt prüft das Programm, ob ein Zugriff auf eine Diskette ab Track 0 (erster Teil der Bootsektoren) erfolgt. Dies ist beim gleich folgenden Booten und auch beim normalen Einlegen einer Diskette der Fall. Auch der Befehl LOADWB greift auf die Bootsektoren zu. Wird VirusControl ohne »-t« aufgerufen, erfolgt auch hier eine Überprüfung durch VirusControl.

Der Amiga bootet nur dann von einer Diskette, wenn folgende zwei Bedingungen erfüllt sind: Die Diskette muß die DOS-Kennung besitzen und die Bootblock-Checksumme muß stimmen. Ist der Bootblock der Diskette kein Standard-Bootblock, handelt es sich um eine »verdächtige Disk«, da es ein Bootblock-Virus sein kann. Es kann sich aber auch um ein harmloses Boot-Intro (z. B. eine Grafik von Bootgirl, AMIGA-Magazin, Ausgabe 3/88) oder ein selbststartendes Spiel handeln.

Wenn der Amiga von einer Diskette booten will, prüft VirusControl, ob die eingelegte Boot-Diskette verdächtig ist. Wenn ja, erscheint ein Dialogfeld (Requester), das eine gefahrlose Bootsimulation anbietet, oder aber die echte Disk-Boot-Routine. Im ersten Fall wird das Programm in den Bootsektoren nicht durchgeführt. Wird die zweite Möglichkeit gewählt, erscheint ein weiteres Dialogfeld, das auf die Gefahr des Aktivierens von Bootblock-Viren durch das »echte« Booten hinweist. Es gibt Spiele, bei denen der Bootblock unbedingt abgearbeitet werden muß, damit das Spiel läuft. In diesem Fall ist es nötig, das Programm in den Bootblocks laufen zu lassen. Wählen Sie diese Möglichkeit jedoch nur, wenn Sie sicher sind, daß es sich nicht um einen Virus handelt!

Wird während der Arbeit mit dem Amiga eine verdächtige Diskette eingelegt, wird dieses durch das Erscheinen eines Kontrollfensters (siehe Bild 1) deutlich gemacht. Hält man während des Disk-Einlegens die rechte Alt-Taste gedrückt, wird die Diskette nicht durch VirusControl überprüft. Drückt man während des Disk-Einlegens die linke Alt-Taste, erscheint auch bei nicht verdächtigen Disketten das Kontrollfenster.

Falls Sie nur über ein Laufwerk verfügen, ist es notwendig, das Kontrollfenster auf andere Art zu öffnen. Rufen Sie VirusControl im CLI oder von der Workbench auf und drücken Sie sofort danach die linke Alt-Taste. Dadurch ist die Diskette im Laufwerk nicht blockiert und Sie können mit VirusControl diese Diskette behandeln.

Kann aufgrund von Speichermangel das Kontrollfenster nicht geöffnet werden, zeigt VirusControl dies durch eine kurze Schwarzfärbung des Bildschirms an. Das Kontrollfenster bietet eine Vielzahl nützlicher Fähigkeiten an, die durch Anklicken des entsprechenden Schalters (Gadgets) ausgewählt werden.

Das Anklicken des Schließ Schalters (Close Gadget, links oben im Kontrollfenster) beendet die Arbeit mit der eingelegten Disk. Erst danach darf eine andere Diskette eingelegt werden. Solange der Schließschalter nicht betätigt wurde, darf die Diskette lediglich zum Ändern des Schreibschutzes entnommen werden. Mit den beiden Schaltern rechts oben, können Sie das Kontrollfenster vor oder hinter die anderen Fenster auf der Workbench legen.

Beende VirusControl

Das Anklicken von »Beende VirusControl« entfernt VirusControl aus dem Speicher. Danach kann die Diskette entnommen werden.

Kill System-Virus

Dieser Programmteil versucht, einen aktiven Virus zu eliminieren, indem unter anderem verschiedene Systemvektoren restau-



Hatten Sie auch schon Ärger mit Viren? Mit »VirusControl« ist die Zeit Sorgen vorbei. Sowohl block- als auch Link-Viren das Programm.

GSTRUPP

riert werden. Es ist allerdings nicht garantiert, daß der Amiga nach dem Versuch richtig weiterarbeitet.

Bootblock -> Schirm

Ein Anklicken von »Bootblock -> Schirm« zeigt den gesamten Bootblock (1024 Byte) als ASCII-Text an. An verräterischen Texten können Sie die meisten Viren erkennen.

Bootblock -> File

Schreibt den gesamten Bootblock(1024 Byte) in eine Datei, deren Name im Texteingabefeld (Stringgadget, ganz unten im Kontrollfenster) eingegeben wird.

File -> Bootblock

Ein Anklicken von »File -> Bootblock« überschreibt den Bootblock mit einer Datei, deren Name im Texteingabefeld eingetragen ist. Mit dieser Möglichkeit, den Bootblock in ein File zu speichern, und dieses File auch wieder auf den Bootblock zu schreiben, können Sie von allen wichtigen Disketten eine Kopie des Bootblocks anfertigen, und somit notfalls den Original-Bootblock wiederherstellen und wieder mit der Diskette arbeiten.

Standard-Bootblock

Haben Sie einen Virus entdeckt, wollen Sie ihn auch beseitigen. Durch das Anklicken von »Standard-Bootblock« wird der Original-Amiga-Bootblock auf Diskette geschrieben. Das entspricht dem Befehl INSTALL.

NoFastMem-Bootblock

Durch Anwählen dieses Schalters schreibt VirusControl eine Bootblock-Routine auf Diskette, die das Fast-Memory schon zu Beginn des Boot-Vorgangs abschaltet. Der Rechner verhält sich so, als ob er nur Chip-Memory besitzen würde. Ein blaues Farbsignal zeigt die NoFastMem-Routine an. Hält man während des Resets

die linke Maustaste gedrückt, wird die NoFastMem-Routine nicht ausgeführt.

f.Checksum -> NoBoot

Schreibt eine falsche Bootblock-Checksumme auf die Diskette, wodurch von der Diskette nicht mehr gebootet werden kann. Ein eventueller Bootblock-Virus ist dadurch (vorübergehend) unschädlich gemacht.

r.Checksum -> NoBoot

Ist das Gegenstück zu »f.Checksum -> NoBoot«, es korrigiert die Checksumme wieder. Dadurch kann mit der Diskette wieder gebootet werden, und somit auch ein eventueller Virus aktiviert werden. Die beiden Routinen ändern nur die Bootblock-Checksumme. Der Bootblock-Inhalt bleibt erhalten.

In S:NoWarning-File

Wie schon erwähnt, gibt es ungefährliche Bootblöcke, die VirusControl als »verdächtig« einstuft. VirusControl wäre unkomfortabel, wenn in diesem Fall jedesmal das Kontrollfenster erscheinen würde. Mit diesem Schalter können Sie diesen Vorgang verhindern. Wenn Sie eine Diskette einlegen, von der Sie GARANTIERTE wissen, daß sich kein Virus darauf befindet, sondern ein ungefährlicher Bootblock, klicken Sie diesen Schalter an. Der verdächtige Bootblock wird nun in der Datei »s:NoWarning« gespeichert. Existiert s:NoWarning, hängt VirusControl den Bootblock an das Dateifeld an. Andernfalls wird S:NoWarning zuerst erstellt. Sollte der

Wichtige Hinweise

Beim Starten folgender Programme öffnet VirusControl ein Warn-Gadget. Sie können es ignorieren, da diese Programme die folgenden Vektoren legal nutzen:

Programm:	Vektoren:
Setpatch	Alert AllocEntry UserState ab V 34,27
Xoper	Switch
CPr	AddTask RemTask OpenDevice
NoFastMem	AllocMem AvailMem
ALF	KickMemPtr
Autoboot-Controller	KickTagPtr KickChecksum

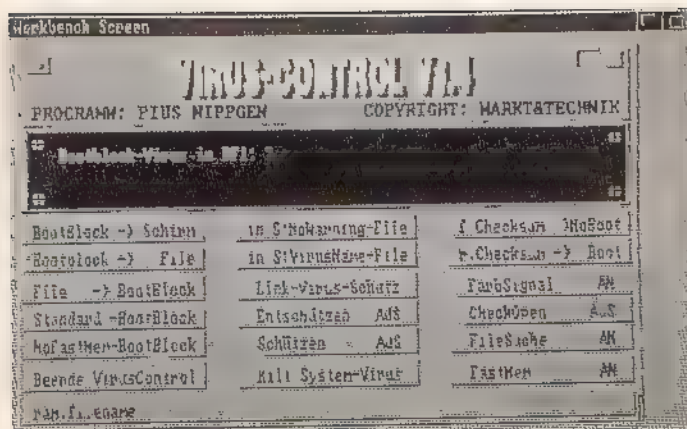


Bild 1. Das Kontrollfenster von »VirusControl« erscheint bei »verdächtigen« Disketten

zu speichernde Bootblock bereits in s:NoWarning vorhanden sein, erscheint ein Requester, der die Möglichkeit anbietet, den Bootblock aus der Datei zu entfernen. Legen Sie nun später eine verdächtige Diskette ein, deren Bootblock in s:NoWarning vorhanden ist, blitzt der Bildschirm nur kurz rot auf. Das Kontrollfenster wird nicht geöffnet. Noch eine Warnung: Übernehmen Sie diese Datei NIE von jemand anderem. Vielleicht will er Sie ärgern!

In S:VirusName-File

Haben Sie einen Virus identifiziert, können Sie in die Datei »s:VirusName« eintragen lassen. Es wird zusätzlich ein Name für den Bootblock gespeichert. Dieser Name wird wieder mittels des String-Gadgets eingegeben. Existiert s:VirusName, wird der Name und der Bootblock an das Datelende angehängt, andernfalls wird s:VirusName zuerst erstellt. Sollte der abzuspeichernde Bootblock bereits in s:VirusName vorhanden sein, erscheinen Requester, die folgende Möglichkeiten anbieten: Man kann den Bootblock und Namen aus s:VirusName entfernen. Man kann den alten Namen durch den neuen Namen ersetzen. Legen Sie nun eine Diskette ein, deren Bootblock in S:VirusName gespeichert ist, wird im Kontrollfenster der Virusname ausgegeben. Mit dieser Methode sind Sie in der Lage, VirusControl ständig auf den neuesten Stand zu bringen.

Wenn eine verdächtige Diskette eingelegt wird, durchsucht VirusControl S:NoWarning und danach S:VirusName. Legen Sie eine Diskette bei gedrückter linker Alt-Taste ein, wird nur s:VirusName durchsucht, da dann immer das Kontrollfenster erscheint. Man kann »s:« mittels ASSIGN auf ein schnelleres Medium umlenken, da das Durchsuchen von 100 KByte (100 Bootblöcke) auf Diskette ca. 10 Sekunden dauert. In RAM:, VD0: oder DH0: benötigt VirusControl dafür nur noch ca. 2 Sekunden. Wenn Sie s: auf die RAM-Disk legen, vergessen Sie nicht, die Dateien am Ende wieder dauerhaft zu speichern. Das Durchsuchen der Files unterbleibt, wenn Sie VirusControl mit dem Parameter »-f« starten. Sind die Dateien

Den Dateizugriff im Griff

in s: nicht vorhanden, unterbleibt zwangsläufig ein Durchsuchen. Alle tiefergehenden Funktionen wie Bootblock beschreiben, VirusControl entfernen oder System-Virus entfernen, sind zusätzlich durch weitere Requester gegen unabsichtliche Ausführung gesichert. Weiterhin zeigen Requester schreibgeschützte Disketten an. Ein Fehler beim Öffnen einer Datei wird ebenfalls durch Requester angezeigt. Da bis zu vier Laufwerke möglich sind, kann VirusControl bis zu vier Kontrollfenster gleichzeitig bearbeiten. Diese sind daher mit dem dazugehörigen Laufwerk gekennzeichnet.

Ein weiß-blaues Farbsignal signalisiert bei jeder neu eingelegten Diskette (oder beim LOADWB-Befehl) oder beim erneuten Starten von VirusControl das korrekte Arbeiten von VirusControl. Dieses optische Zeichen kann jedoch abgeschaltet werden, indem man VirusControl zusätzlich mit »-c« aufruft. Um das Kontrollfenster besser zur Geltung zu bringen, werden vorübergehend neue Workbench-Farben gesetzt und am oberen und unteren Bildschirmrand ein blauer Rahmen verwendet.

Beim Starten von VirusControl überprüft das Programm die eingelegten Disketten. Bei Angabe von »-t« wird dieses unterlassen.

Geben Sie den Parameter »-o« ein, bewirkt dies, daß vor jedem Filezugriff ein Requester erscheint. Mit ihm können Sie den Zugriff erlauben oder verbieten. Hält man jedoch die Ctrl-Taste gedrückt, wird direkt zur Original-Routine gesprungen. Diese Dateizugriffskontrolle kann sehr nützlich für das Erkennen von Link-Viren sein. Link-Viren breiten sich nicht über den Bootblock aus, sondern sie hängen sich an bestehende Dateien. Sie verlängern also diese Dateien und werden durch das Aufrufen derselben aktiviert. Wenn sich nun ein Link-Virus an ein Programm hängen will, kann der Filezugriff mittels des Requesters erkannt und verhindert werden. Diese Option ist nur zum Suchen gedacht, wenn Sie vermuten, daß sich ein Link-Virus eingeschlichen hat.

Link-Virus-Schutz

VirusControl bietet einen weiteren äußerst wirksamen Schutz vor Link-Viren. Das Datei-Schutz-System wird mit dem Gadget »Link-Virus-Schutz« aufgerufen. Es wird die im String-Gadget angegebene Datei nach der VirusControl-Schutz-Kennung durchsucht. Weist das File noch keine Schutz-Kennung auf, schützt VirusControl (sofern es sich um ein Programm handelt) die Datei automatisch. Die Schutzfunktion läßt sich auch auf ganze Unterverzeichnisse anwenden. Wenn man VirusControl mit »-s« aufruft, oder wenn man »Schützen An« durch Anklicken setzt, wird eine Datei erst nach Rückfrage geschützt. Dateien, die nur Daten enthalten, werden generell nur nach Rückfrage geschützt, da es nicht sinnvoll ist, z. B. eine Textdatei zu schützen. Anstatt eines einzelnen Dateinamens sind auch Verzeichnis-Angaben erlaubt. Man kann also komfortabel das komplette C-Verzeichnis schützen lassen. Das automatische Durcharbeiten eines kompletten Verzeichnisses kann jedoch jederzeit durch das gleichzeitige Drücken von linker und rechter Shift-Taste gestoppt werden.

Weist eine Datei bereits die Schutzkennung auf, wird nun geprüft, ob die Datei verändert wurde. Ist dies der Fall, erscheint ein Requester, der auf den vermutlichen Link-Virus-Befall hinweist. Mittels des Requesters ist es nun möglich, den Schutz der Datei wieder zu entfernen. Beim nächsten Durchgang wird diese Datei mit einer neuen Kennung versehen. Mit File-Entschützen kann aber ein Link-Virus meist nicht entfernt werden. Leider sind wegen des flexiblen Aufbaus für Programmdateien viele Link-Viren-Typen denkbar. Es ist also unmöglich, ein generelles Programm zum Entfernen von Link-Viren zu schreiben. Vielmehr muß man für jeden Link-Virus gezielt ein spezielles Programm erstellen.

Es empfiehlt sich also folgende Vorgehensweise, um sich vor Link-Viren zu schützen:

- Sicherheitskopien der Originaldateien anfertigen
- Originaldateien schützen (mit »Link-Virus-Schutz«)
- Regelmäßig prüfen, ob die Dateien noch unverändert sind.
- Wird eine Veränderung festgestellt, deutet dies auf einen Link-Virus hin. Man entfernt nun diesen Link-Virus einfach dadurch, indem man das verdächtige Programm mit dem Original-Programm überschreibt. Dieses neue Programm sollte nun sofort wieder geschützt werden.

Durch das Anklicken von »Link-Virus-Schutz« veranlassen Sie VirusControl, noch ungeschützte Dateien zu schützen und bereits geschützte auf Veränderung zu überprüfen. Dies geschieht weitgehend automatisch. Wird jedoch »Schützen An« gesetzt, muß das »File schützen« extra bestätigt werden. Es empfiehlt sich, beim erstenmal die Verzeichnisse automatisch schützen zu lassen. Danach sollte man aber »Schützen An« setzen, denn dadurch wird man in Zukunft extra mittels Requester darauf aufmerksam gemacht, wenn eine neue Datei im Verzeichnis vorliegt. Ein Virus könnte durchaus ein gefährliches Programm erstellen, bzw. ein vorliegendes Programm einfach überschreiben. Dieses kann nun nicht mehr übersehen werden. Wenn eine Veränderung festgestellt wird, muß auch diese wegen eines möglichen Link-Virus-Befalls extra bestätigt werden. Wenn Sie VirusControl mit »-e« aufrufen oder »Entschützen An« durch Anklicken setzen, bietet VirusControl die Möglichkeit, die Datei zu entschützen, wodurch das Original-Programm wieder zur Verfügung steht. VirusControl zeigt alle File-Schutz-Vorgänge am Bildschirm an. Ferner wird ein dauerhaftes Protokoll in der Datei »s:VCprotokoll« angelegt.

Man sollte die Verzeichnisse »C:«, »L:«, »LIBS:«, und »DEVS:« schützen, da hier ausführbare Programme enthalten sind, die entweder von Benutzer selbst oder vom Betriebssystem aufgerufen werden. Bei diesem Aufruf könnte dann ein Link-Virus aktiviert werden. Info-Dateien werden generell nicht geschützt.

Die Kurzanleitung zu VirusControl kann durch erneutes Starten

von VirusControl abgerufen werden. Hierbei können auch die Parameter (»C«, »f«, »t«, »O«, »m«, »e« und »s«) neu gesetzt werden. Geben Sie keine Parameter an, verwendet VirusControl die Standard-Einstellung:

- Farbe AN — ChipMemFirst AUS
- FileSuche AN — Entschützen AUS
- Kein Test bei Datei öffnen — Schützen AUS

Die Parameter können auch mittels der entsprechenden Gadgets (FarbSignal, CheckOpen, FileSuche, ChipMemFirst, Entschützen, Schützen) neu gesetzt werden.

Kontrolle der Systemvektoren

VirusControl kann also mit Sicherheit das Eindringen von Bootblock-Viren verhindern, da es verdächtige Disketten anzeigt und eine gefahrlose Bootsimulation anbietet. Ferner zeigt VirusControl mittels des Dateischutz-Systems Link-Viren an.

Außer allen bisher genannten Maßnahmen installiert VirusControl einen separaten Task zur Überprüfung der Systemvektoren. Wird eine Veränderung festgestellt, zeigt VirusControl mittels Requester die jeweilige Veränderung an. Nach Anklicken von »ENDE« wird normal weitergearbeitet. Ein Anklicken von »VIRUS-ENTFERN-VERSUCH« bewirkt, daß einige Vektoren auf den Standardwert gesetzt werden. Diese Option kann jedoch zu einem Absturz führen, da VirusControl mit dem Virus kollidiert. Es ist daher ratsam, den Virus durch Ausschalten des Amiga zu entfernen, denn nur diese Methode kann wirklich sicher sein. Sollte man es einem Virus erlauben haben, sich im System zu installieren, indem man im »KILL SYSTEM VIRUS«-Requester »ENDE« anklickt, kann man dennoch später jederzeit versuchen, den Virus zu entfernen, bzw. VirusControl wieder voll zu installieren. Dieses ist durch den separat laufenden VirusControl-Task möglich. Durch gleichzeitiges Drücken von linker und rechter Alt-Taste wird der »KILL SYSTEM VIRUS«-Requester aufgerufen.

Änderungen an den Vektoren können aber auch von seriösen Programmen herrühren. So »verbiegt« das Disketten-Cache-Programm FACC ebenfalls den BeginIO-Vektor der trackdisk.device-Struktur. Die resetfeste RAM-Disk RAD: benutzt die KICK-Vektoren. Der Setpatch-Befehl verbiegt den Alert-Vektor und AllocEntry-Vektor.

Um herauszufinden, ob es sich bei einem verdächtigen Bootblock wirklich um einen Virus handelt, kann man folgendermaßen vorgehen: Man bootet mit installiertem VirusControl von der verdächtigen Disk, aktiviert also bewußt einen eventuellen Virus. Alle eingelegten Disketten sollten schreibgeschützt sein, damit sich der eventuelle Virus nicht verbreiten kann. Handelt es sich um einen Virus, kann man dies an folgenden Umständen erkennen:

- Der Vektor-Kontroll-Task weist mittels Requester auf veränderte Vektoren hin
- Ein Schreibzugriff auf den Bootblock wird durch einen Requester angezeigt.
- Kein weiß-blaues Farb-Signal beim Einlegen einer neuen Disk, obwohl das Farb-Signal eingestellt ist.

Viren bringen Computer in den Ruf der Unzuverlässigkeit. Das kann nicht im Interesse der Amiga-Benutzer sein. Es hat sich schon gezeigt, daß Viren-Programmierer auch in der Lage sind, »vernünftige« Programme zu entwickeln

Es wäre schade, wenn folgender Ausspruch von Jim Mackraz (Amiga Software-Entwickler, arbeitet zur Zeit am Intuition-Teil der Kickstart 1.4) noch lange Gültigkeit besitzt: "Die beste Software aus Deutschland kommt meistens im Bootblock!"

- Beim Disk-Einlegen erscheint trotz gedrückter linker Alt-Taste kein Kontrollfenster
- Bei einem Reset tritt kein weiß-blaues Farb-Signal auf.
- Beim Aufruf des VirusControl-Programmes wird eine Meldung ausgegeben, die auf die vermutliche Virus-Infektion hinweist.

Da sich der Amiga flexibel programmieren läßt, gibt es auch für ein Virus-Programm viele Möglichkeiten. Es werden daher nicht alle eben genannten Effekte auftreten. Es kann auch zu Kollisionen zwischen VirusControl und dem Virus kommen, die zum Absturz führen. Anschließend muß der Amiga ausgeschaltet werden, da dies die einzige sichere Methode ist, um einen aktiven Virus zu entfernen.

Die einzige sichere Methode, um einen Bootblock-Virus zu identifizieren, ist zugleich auch die gefährlichste. Man bootet ohne VirusControl von der verdächtigen Disk, und legt eine nicht schreibgeschützte, nicht infizierte Diskette ein. Nun schaltet man den Amiga aus, wartet ca. 20 Sekunden, schaltet den Amiga wieder ein, bootet von einer sauberen Disk, startet VirusControl und legt die nicht schreibgeschützte Diskette ein. Wenn die Diskette nun verdächtig geworden ist, handelt es sich um einen Virus. Die Diskette sollte sofort »überschrieben« werden, damit man später nicht versehentlich von dieser Diskette bootet, und somit den Virus aktiviert und dadurch weiterverbreitet.

Da VirusControl nur wenig Rechenzeit in Anspruch nimmt, ist es empfehlenswert, das Programm immer laufen zu lassen. Es beeinträchtigt Ihre Arbeit am Computer nicht, bringt Ihnen aber eine hohe Sicherheit gegen Viren jeglicher Art.

Die Gefahr der »Infektion« ist auf PC- und Macintosh-Systemen höher als auf dem Amiga. Damit dies so bleibt und die Gefahr noch geringer wird, sollten Sie VirusControl unbedingt einsetzen. Denn auf keinem anderen Computer gibt es ein so leistungsfähiges Programm zum Aufspüren dieser lästigen Viren. P. Nippgen/R. Beaupoil/kn

Programmname: VirusControl_Gen	
Computer: A500, A1000, A2000 mit Kickstart 1.2 und 1.3	
Sprache: Amiga-Basic 1.2	
Bemerkung: Erzeugt lauffähiges Programm	
Programmautor: Pius Nippgen ----- 1 OmQ REM Generiert lauffähiges Progr =====	
2 LI ON ERROR GOTO Fehler	29 Sv CLOSE
3 bh CLS	30 Na KILL "DFO:s/startup-sequences.in fo"
4 VT PRINT "Bitte Workbench-Diskette in DFO: einlegen!"	31 IL KILL "DFO:s/Oldss"
5 b5 a\$=""	32 KU PRINT "Booten Sie nun von der D iskette in DFO: I"
6 wm WHILE a\$="" a\$=INKEYS WEND	33 HC END
7 9v OPEN "DFO:e/VirusControl" FOR C UTPUT AS 1	34 Ts Fehler:
8 FM READ anz	35 W2 PRINT "Fehler: "ERR
9 ae FOR i=1 TO anz	36 e ON ERROR GOTO 0
10 7r1 READ h\$	37 LGO END
11 2r2 wert1=ASC(LEFT\$(h\$,1))	38 Jx Werte:
12 fT IF wert1>64 THEN wert1=wert1- 87 ELSE wert1=wert1-48	39 zd DATA 13592
13 JM wert1=wert1*16	40 OQ DATA 00,00,03,f3,00,00,00,00,00
14 Bg wert2=ASC(RIGHT\$(h\$,1))	41 xY DATA 00,00,03,00,00,00,00,00,00
15 Ot IF wert2>64 THEN wert2=wert2- 87 ELSE wert2=wert2-48	42 gt DATA 00,02,00,00,00,d8,00,00,00
16 Tm wert=wert1+wert2	43 F8 DATA 5f,00,00,1e,3d,00,00,03,e9
17 DK PRINT #1,CHR\$(wert);	44 fo DATA 00,00,00,d8,48,e7,ff,fe,4b
18 NSQ NEXT	45 WC DATA fe,03,38,41,fe,ff,f2,22,50
19 7r CLOSE 1	46 CK DATA d3,c9,d3,c9,24,51,58,89,2b
20 ku KILL "DFO:e/VirusControl.info"	47 3I DATA 49,00,14,d3,ca,d5,ca,58,8e
21 ae NAME "DFO:s/startup-sequences" A S "DFO s/Oldss"	48 xD DATA 2b,4e,00,18,48,e7,ff,fe,61
22 ZQ OPEN "DFO:s/Oldss" FOR INPUT AS #1	49 Ia DATA 00,01,b0,40,d7,7f,ff,22,6d
23 ut OPEN "DFO:s/startup-sequences" F OR OUTPUT AS #2	50 uz DATA 00,14,51,89,20,11,2c,78,00
24 iq PRINT #2,"VirusControl"	51 x4 DATA 04,4e,ae,ff,2e,26,6d,00,18
25 qq WHILE NOT EOF(1)	52 uf DATA 50,8b,20,1b,2e,00,e7,88,22
26 M82 LINE INPUT #1,a\$	53 KO DATA 3c,00,01,00,00,4e,ae,ff,3a
27 UW PRINT #2,a\$	54 cm DATA 2b,40,00,0c,50,8b,20,07,53
28 thQ WEND	55 GI DATA 80,22,6d,00,0c,22,1b,58,89
	56 FV DATA 54,81,22,c1,51,58,ff,f6,2f
	57 9u DATA 0b,2c,07,53,86,26,6d,00,0c
	58 F8 DATA 28,4b,4a,9b,22,3c,00,01,00
	59 y2 DATA 00,20,1b,58,8b,08,00,00,1e
	60 H2 DATA 67,04,08,c1,00,01,e5,88,4e

Listing. »VirusControl« macht Schluß mit Viren aller Art ►



61 wt DATA ae,ff,3e,28,e0,58,8e,51,ee
62 xu DATA ff,e0,26,3f,7c,00,7a,00,0c
63 J1 DATA 6b,03,e9,00,02,67,00,00,9a
64 5t DATA 0c,6b,03,ea,00,02,67,00,00
65 NT DATA 90,0a,6b,03,eb,00,02,67,00
66 h1 DATA 00,a8,0c,6b,03,ea,00,02,67
67 QK DATA 00,00,b0,0c,6b,03,f2,00,02
68 11 DATA 4e,71,4a,9b,7a,00,52,86,be
69 YP DATA 86,0e,c6,2f,07,43,fa,ff,18
70 Ry DATA 20,6d,00,0e,20,10,58,80,e4
71 Jq DATA 88,22,80,52,87,53,f7,e7,1a
72 Wy DATA 22,50,20,28,00,04,e5,88,22
73 0s DATA 80,20,28,00,08,58,80,e4,88
74 0b DATA 23,40,00,04,50,88,60,e2,2e
75 tk DATA 1f,42,e9,00,04,43,fa,00,2a
76 Pu DATA 20,6d,00,0e,20,50,50,88,22
77 tW DATA 88,22,6d,00,0c,20,07,e7,88
78 JM DATA 4e,ae,ff,2e,22,6d,00,18,51
79 mf DATA 89,20,11,4e,ae,ff,2e,4c,df
80 Yg DATA 7f,ff,4e,f9,00,00,00,00,61
81 Lj DATA 28,4a,9b,20,1b,e5,88,20,4b
82 bx DATA 87,c0,22,6d,00,0c,22,06,e7
83 89 DATA 89,22,71,18,00,50,89,4e,ae
84 gd DATA fd,90,60,00,ff,3e,61,06,50
85 1j DATA 8b,60,00,ff,36,4a,85,67,02
86 Qa DATA 52,86,7a,01,4e,75,4a,9b,20
87 Y1 DATA 6d,00,0c,20,06,e7,88,22,70
88 2S DATA 08,00,50,89,20,1b,67,24,22
89 LB DATA 1b,e7,89,20,6d,00,0c,28,70
90 Qc DATA 18,00,50,8c,53,80,20,4c,24
91 Jt DATA 49,22,1b,d5,c1,22,12,d1,c1
92 Rv DATA 2a,88,51,c8,ff,f0,60,38,60
93 OT DATA 70,fe,f0,48,e7,ff,fe,4b,fa
94 mb DATA 01,80,41,fa,ff,f2,22,50,d3
95 8F DATA c9,d3,c9,07,88,2b,49,00,14
96 28 DATA 2b,7c,00,00,50,00,00,18,61
97 qD DATA 0c,4e,b9,00,00,00,00,4e,df
98 Te DATA 7f,ff,4e,75,20,6d,00,14,2a
99 rx DATA 18,22,6d,00,18,d3,c5,d1,d0
100 3o DATA 91,fc,00,00,00,04,4a,60,6a
101 5Q DATA 02,53,88,10,20,e3,08,66,04
102 29 DATA 10,20,e3,10,64,5a,42,41,e3
103 du DATA 08,66,04,10,20,e3,10,64,42
104 ye DATA 47,fa,00,38,76,03,42,41,14
105 ud DATA 33,30,00,48,82,78,ff,e5,60
106 is DATA 46,44,53,42,e3,08,66,04,10
107 ei DATA 20,e3,10,e3,51,51,ca,ff,f4
108 at DATA 4a,43,67,06,b8,41,5b,cb,ff
109 mW DATA da,14,33,30,04,48,82,d2,42
110 mo DATA 60,00,00,0a,0a,03,02,02,0e
111 uD DATA 07,04,01,13,20,33,c8,00,df
112 W1 DATA 1f,80,51,e9,ff,f0,47,fa,00
113 De DATA d6,26,6b,00,14,50,8b,1b,cb
114 zd DATA 6f,00,00,c8,47,fa,00,3a,74
115 KN DATA 03,e3,08,66,04,10,20,e3,10
116 F7 DATA 64,04,51,ca,ff,f4,42,41,52
117 nr DATA 42,16,33,20,00,67,12,48,83
118 Pv DATA 53,43,e3,08,66,04,10,20,e3
119 ge DATA 10,e3,51,51,cb,ff,f4,16,33
120 RV DATA 20,05,48,83,d3,60,00,00,00
121 gw DATA 0c,0a,02,01,00,00,0a,06,04
122 Jz DATA 03,02,0c,41,00,02,67,42,47
123 ze DATA fa,00,34,76,01,e3,08,66,04
124 Du DATA 10,20,e3,10,64,04,51,cb,ff
125 OT DATA f4,52,43,42,42,18,33,30,00
126 AK DATA 48,84,e3,08,66,04,10,20,e3
127 tN DATA 10,e3,52,51,ce,ff,f4,e3,4b
128 Ru DATA d4,73,30,04,60,00,00,32,0b
129 PU DATA 04,07,00,01,20,00,00,00,20
130 Lk DATA 00,00,42,42,76,05,42,44,e3
131 LK DATA 08,66,04,10,20,e3,10,64,04
132 WQ DATA 76,08,78,40,e3,08,66,04,10
133 FG DATA 20,e3,10,e3,52,51,cb,ff,f4
134 Ze DATA 84,44,45,f1,20,00,46,c1,d5
135 ZO DATA c1,53,41,13,22,51,e9,ff,fe
136 12 DATA 60,00,fe,e8,4e,75,00,00,00
137 Jr DATA 00,00,00,00,00,00,00,00,00
138 Xs DATA 00,00,00,00,00,00,00,00,00
139 Et DATA 00,00,00,00,00,00,00,00,00
140 o4 DATA 00,00,a0,5a,00,00,03,ea,00
141 MP DATA 00,0a,5f,00,00,68,fa,00,00
142 2s DATA 31,7c,b2,f3,1a,cc,02,7d,04
143 2P DATA 73,45,29,28,a6,a6,03,82,00
144 14 DATA 19,f1,34,33,cc,4e,f9,01,a8

145 L1 DATA 23,26,a6,f6,de,40,4a,a9,00
146 OE DATA ac,66,4e,2f,00,79,fe,61,80
147 S6 DATA 22,5f,41,e9,00,5c,ee,f7,8c
148 LX DATA d1,04,be,24,40,24,6a,00,dd
149 61 DATA 13,12,23,ea,88,03,70,31,ba
150 Ob DATA b8,1e,d5,a6,82,4b,90,d3,d3
151 DB DATA 68,35,03,ed,73,e2,54,97,20
152 HE DATA 60,12,7b,1f,fa,d0,ab,c4,d6
153 yV DATA d7,b6,ed,14,2a,e4,f9,fc,9b
154 Ww DATA c0,8d,33,b8,3d,4c,66,28,75
155 yd DATA 9f,d0,aa,c1,09,1e,d6,64,92
156 qY DATA b9,0c,38,2e,17,c4,48,9d,17
157 S1 DATA d0,b3,01,38,23,4d,73,3a,4c
158 w1 DATA 69,cc,0a,c7,19,72,59,0e,07
159 po DATA e0,c7,1c,99,31,02,86,b3,46
160 Me DATA 6d,75,e0,77,85,6e,1c,72,6e
161 o1 DATA 22,b5,79,e0,30,7b,cd,a8,dd
162 xH DATA 26,6a,f3,80,5c,f7,9a,8a,cb
163 0e DATA 02,2a,67,a4,46,60,c0,88,1b
164 pM DATA 21,6c,41,ee,01,5e,43,fa,08
165 SW DATA 78,c7,1f,30,ce,20,40,41,eb
166 nY DATA e5,bf,23,c8,6c,34,e0,c0,90
167 e3 DATA e0,26,64,18,98,7b,e7,2c,d6
168 zE DATA 72,d6,10,b4,59,73,66,a9,be
169 aE DATA 3a,42,10,0c,26,7a,16,d8,f4
170 nD DATA cd,1c,96,5d,66,4a,0f,8e,26
171 k2 DATA ea,c8,70,7f,d1,8b,5b,df,a6
172 rP DATA ea,e4,f8,ac,2a,dc,0c,a5,06
173 9P DATA 4a,99,00,18,d9,76,04,86,4a
174 Ds DATA b6,a3,7c,66,78,88,91,08,98
175 KY DATA 67,52,c1,c5,67,66,08,9c,0d
176 KX DATA 30,06,60,00,16,ac,fa,a5,88
177 uD DATA 22,3a,07,d6,67,04,3e,f9,8c
178 no DATA 2d,9a,b3,2f,ec,03,83,fa,2c
179 Ha DATA 8f,ce,14,6d,48,6a,16,0c,20
180 HS DATA 20,3b,aa,58,a6,61,00,2d,7e
181 iB DATA 94,9f,9c,7c,d5,07,e8,80,bf
182 2x DATA 6c,86,22,7a,32,be,20,09,7d
183 Wx DATA 33,74,52,28,03,77,41,fa,35
184 PT DATA e8,64,00,5a,00,4d,ab,00,69
185 LT DATA 18,02,ac,b9,00,02,06,1c,5a
186 Sn DATA 01,52,db,60,00,6d,86,00,35
187 k6 DATA 83,00,ba,1c,b6,4a,e0,ee,26
188 zM DATA ea,35,a8,e0,01,58,eb,60,00
189 Yh DATA 4d,8b,00,35,84,01,56,1c,54
190 Zh DATA 2a,01,ec,91,b0,01,ae,1c,54
191 iu DATA 15,02,ed,02,a2,25,02,b4,9e
192 rr DATA 02,a6,02,d0,48,03,9a,04,03
193 Hq DATA 6b,35,03,a8,03,2d,0d,03,b4
194 nQ DATA 8a,03,46,04,d0,ba,04,32,58
195 cG DATA 0e,2a,98,05,06,05,d3,4e,00
196 cT DATA 9a,06,42,4b,06,69,06,04,0a
197 eP DATA 2c,07,55,04,44,f0,04,68,11
198 hx DATA 04,fe,39,a8,28,05,5a,06,42
199 or DATA a3,61,06,2d,06,e6,26,07,15
200 5S DATA 03,07,6b,36,07,4f,07,5e
201 2v DATA 35,9a,e8,a4,f5,c1,07,1a,08
202 vb DATA 4c,13,08,69,17,08,6d,08,6a
203 qK DATA 35,09,10,39,a8,34,09,9a,02
204 xm DATA 09,60,cb,41,8c,0d,2a,0b,09
205 pL DATA d6,24,09,b0,58,0e,dc,aa,09
206 DU DATA 80,69,b0,0a,86,c1,0a,00,72
207 FS DATA 2c,50,83,28,1a,54,1a,0a,50
208 eR DATA 2c,07,7c,35,78,a8,41,25,c2
209 Im DATA 0a,a0,72,cc,50,81,c8,1a,f4
210 SH DATA 34,0a,01,d3,0b,68,05,0b,18
211 Ym DATA 39,44,0a,41,25,0b,d2,0c,0b
212 7d DATA 5a,04,0b,6b,14,0b,ac,a9,0b
213 JC DATA b4,83,0b,b8,1c,e4,54,0b,01
214 03 DATA 93,0c,69,08,0c,08,39,1c,a8
215 AV DATA 41,75,10,3e,90,83,48,1a,4c
216 7F DATA 04,a0,7a,10,68,0f,10,8e,39
217 v9 DATA a8,12,10,90,72,50,59,10,34
218 yP DATA 8b,10,46,10,40,20,10,ee,72
219 Yx DATA 50,75,11,b4,0e,11,b6,11,d0
220 LG DATA 2e,11,37,72,f0,52,83,4a,10
221 4R DATA 2a,21,07,c4,35,a8,14,0f,de
222 VF DATA 72,50,ed,0f,34,85,0f,b2,1c
223 K7 DATA 54,35,0f,4d,0f,8b,21,07,40
224 OI DATA 35,a8,14,0f,5a,72,14,50,83
225 Uq DATA 4a,0f,2e,21,07,e8,b5,ae,14
226 UO DATA 0f,02,72,bc,50,83,4a,0a,a1
227 xP DATA ad,0e,90,e4,a0,52,0e,aa,e8
228 Et DATA 41,85,0e,d1,1a,0e,7e,72,38

229 1a DATA 50,83,4a,1a,5c,d4,a0,0c,06
230 nJ DATA 1e,55,0a,26,80,83,e0,1a,f2
231 w7 DATA d4,a0,fa,06,55,0d,45,ab,0d
232 oH DATA ce,c8,41,d5,0d,d0,04,0d,a2
233 pU DATA 72,5c,50,83,4a,0d,76,20,07
234 pB DATA 30,55,0d,04,80,83,a8,1a,ae
235 E4 DATA d4,a0,5a,0c,92,c8,41,9a,0d
236 uy DATA 54,6a,66,50,83,6e,1a,3a,54
237 Kk DATA 0c,42,01,0e,64,6a,68,50,83
238 w1 DATA 7a,1a,84,d4,a0,88,0b,9a,35
239 2M DATA 98,a8,41,a4,0d,a8,6a,b4,50
240 P5 DATA 05,11,c6,38,6c,a8,02,12,80
241 Fc DATA 1c,86,d4,a0,da,0b,d6,55,1d
242 It DATA d2,80,83,1a,1a,22,d4,a0,36
243 oK DATA d1,a8,fd,7e,d6,0b,d2,cb,62
244 UJ DATA 56,97,12,ea,c8,ba,05,7a,50
245 bE DATA 48,00,6c,f8,df,20,09,22,ee
246 QD DATA 20,fc,22,fe,38,2c,2f,5c,52
247 7y DATA e0,99,4d,a0,6a,6a,2f,5c,3f
248 s2 DATA ae,ee,9c,6a,3f,b8,99,61,75
249 nJ DATA 02,9e,26,88,df,3f,e0,66,2e
250 sY DATA 99,40,20,d9,6b,c7,75,ea,d2
251 93 DATA 7a,b6,ef,f4,ca,a0,10,06,38
252 tA DATA 13,67,54,47,ea,2f,d0,c0,58
253 I3 DATA 47,11,67,06,55,88,24,d8,60
254 6x DATA 0c,30,18,5a,c0,7f,c9,24,40
255 8P DATA 31,95,58,88,b5,cb,b5,ea,30
256 TS DATA 3c,12,6b,20,49,f9,c3,f0,44
257 QD DATA 2c,42,18,ec,26,3d,a3,2e,ce
258 dG DATA 18,57,c5,34,5a,15,49,ab,6f
259 VM DATA cc,43,d0,ec,0b,09,c0,97,eb
260 Lu DATA b3,e7,c6,eb,aa,41,81,60,6a
261 G1 DATA 67,44,4f,49,4f,06,8f,6a,98
262 x3 DATA 2b,8a,97,84,00,d9,cc,02,0f
263 zw DATA 56,74,36,c0,40,e8,32,e4,c3
264 IQ DATA 60,42,0d,2b,64,20,36,04,ea
265 pa DATA 32,1f,a2,15,c1,72,52,07,28
266 U4 DATA aa,31,9b,1d,9a,15,41,54,ce
267 ID DATA 8e,9e,23,fc,36,ca,20,ad,06
268 BO DATA 0a,e2,13,fc,4e,1f,33,61,02
269 fy DATA 60,66,0e,c7,ae,a0,5d,fa,67
270 XD DATA 32,9b,56,00,0c,6e,a1,d0,ab
271 qA DATA 11,e2,67,46,ba,ed,e8,9a,da
272 pU DATA 4a,e8,39,92,76,01,b3,cb,f2
273 B1 DATA 22,3a,02,ea,ec,75,dc,1a,28
274 EN DATA 90,85,97,86,b6,b6,70,c3,7b
275 H1 DATA 7c,d4,1f,00,11,72,79,0b,fa
276 k7 DATA da,b0,0d,28,ba,f6,5a,18,57
277 X3 DATA 0c,00,86,74,ba,ff,ff,3f,5e
278 bG DATA 80,1b,09,cc,ae,06,fe,73,a3
279 1a DATA d6,b9,42,9c,96,4a,aa,26,35
280 Ir DATA 83,03,a5,02,aa,73,29,86,15
281 3F DATA 85,fb,be,67,3c,76,57,cc,15
282 OQ DATA 7e,01,67,d5,45,3c,c3,7b,2b
283 zw DATA 9a,00,b2,b7,1e,27,f2,9e,01
284 aQ DATA b2,b7,d4,e6,20,7a,02,12,f9
285 kh DATA 7b,36,20,50,dc,c6,e1,00,d6
286 Nx DATA 99,f3,29,57,84,60,06,4a,fa
287 LS DATA 01,01,01,ec,f9,bd,01,f0,53
288 LO DATA d5,ac,3b,be,6b,f0,66,04,33
289 Kc DATA 6f,f6,35,8f,c4,87,57,67,1a
290 Gc DATA 63,06,76,68,dd,e0,32,56,dc
291 Hv DATA d8,d4,a6,74,66,0a,42,7e,a7
292 un DATA d,c8,75,bf,aa,49,75,5a,28
293 P2 DATA 2e,69,0a,b2,68,01,6d,66,22
294 BX DATA bb,80,74,0c,30,26,e6,4b,4f,fe
295 hD DATA f2,0d,82,c5,21,4a,10,63,08
296 m8 DATA 0d,ce,a6,8a,5a,65,d5,69,73
297 nU DATA 1b,ed,fe,9d,74,b6,79,be,73
298 AK DATA 66,12,5d,72,d5,aa,2a,4e,2f
299 tv DATA d6,52,88,f8,a9,5c,0c,18,00
300 JW DATA 66,66,10,42,29,23,77,23,7c
301 MP DATA fd,c9,6c,3d,26,e6,4b,4f,fe
302 LX DATA 0a,24,7a,01,28,45,ea,1b,8e
303 64 DATA 4e,92,24,5f,fe,00,fe,0f,f4
304 q5 DATA 32,c5,b1,80,e0,05,c8,72,32
305 nD DATA a1,a1,61,2e,91,de,3c,c5,01
306 QG DATA 75,40,82,3a,dd,79,5e,54,4a
307 wB DATA 70,9c,61,46,47,01,61,24,6d
308 Eo DATA 61,20,70,03,61,1c,4a,b9,23


```

309 WD DATA fe,53,b0,c6,67,fc,d9,eb,e8
310 WS DATA aa,3a,37,02,52,f3,ce,2e,ad
311 c4 DATA 2e,fe,1e,2f,00,93,c9,eb,a1
312 Cx DATA 60,f6,ce,54,3a,b8,13,10,3a
313 b0 DATA 9e,d5,ab,bc,52,71,e1,be,b6
314 Xg DATA 72,00,20,1f,7c,05,44,3d,b8
315 Bo DATA 6e,35,82,a2,3a,c0,2d,98,d7
316 7c DATA e5,1f,ad,0e,26,28,68,8a,23
317 12 DATA 4b,00,28,70,02,33,e0,27,1c
318 93 DATA e1,88,23,40,e0,5d,24,2c,d0
319 VE DATA 66,6e,ee,72,04,77,a5,97,14
320 vA DATA 5a,57,8a,b0,e5,09,00,1c,42
321 22 DATA a9,e5,d7,79,31,38,b8,42,d4
322 oD DATA 64,db,f3,fe,3e,22,7a,00,34
323 mI DATA d3,f5,97,6e,02,30,4a,ee,fe
324 qQ DATA 98,7b,60,02,0f,08,c0,03,b6
325 6W DATA fc,bf,ff,c5,eb,19,b5,cd,b0
326 J9 DATA 6d,c6,6d,76,b3,89,a2,70,8b
327 3B DATA 68,67,00,04,36,34,58,32,40
328 wT DATA 47,2f,69,2d,ef,20,28,cc,32
329 dE DATA 8t,6b,1c,cc,31,cc,54,fc,c4
330 Nc DATA 6e,67,2d,73,29,db,58,d6,2d
331 b1 DATA 2d,96,75,eb,64,2e,b3,56,65
332 0H DATA 62,59,e6,20,2d,20,54,e6,77
333 ot DATA fa,00,61,26,67,1e,24,40,20
334 5L DATA 2a,00,3a,04,cd,3f,2c,5d,68
335 fA DATA 97,37,77,2e,00,00,09,ae,22
336 De DATA 00,00,2e,b3,c0,66,02,f6,63
337 DX DATA 58,8f,60,00,fc,ae,f4,f6,36
338 aE DATA 43,fa,0f,3c,7b,c1,da,4a,80
339 se DATA 6e,23,ec,38,04,59,82,07,e0
340 pv DATA 07,f0,b2,66,66,74,13,11,e9
341 3E DATA c2,86,ea,02,de,ed,cc,ee,67
342 MV DATA 25,0d,73,71,6d,41,18,29,90
343 2P DATA 24,8b,71,46,33,e2,40,35,66
344 3x DATA fc,e4,18,88,b3,7c,24,81,73
345 xQ DATA 6e,45,68,45,6b,fc,4c,6c,2e
346 0e DATA 53,79,29,77,6d,a2,7c,73,0e
347 2t DATA 80,00,63,31,61,6b,88,bf,a4
348 nM DATA 8e,36,e8,dc,6d,2f,52,70,29
349 rM DATA 42,73,66,b9,32,23,c1,76,65
350 mQ DATA 36,08,b3,4b,b0,79,98,b5,d2
351 6L DATA 62,7a,b2,93,8c,2b,9f,2f,39
352 3k DATA 97,47,07,70,66,7d,0e,80,ce
353 g6 DATA 14,56,d6,cc,66,65,58,6e,c2
354 ah DATA 72,ee,8d,ba,e6,59,78,2b,4c
355 FZ DATA 38,5b,f1,f4,2e,ca,d3,37,12
356 FF DATA 5c,76,0c,32,2e,64,e4,eb,a3
357 YB DATA 5a,58,5c,9b,8a,19,9d,dc,24
358 DH DATA 34,07,78,8a,70,c3,c0,2a,52
359 ea DATA c2,2d,e7,18,67,8c,6b,2b,11
360 bB DATA 0f,6e,2b,82,2b,20,5e,09,f0
361 G1 DATA 22,30,18,58,19,b2,c5,3d,c3
362 Qq DATA d2,21,b2,8b,64,6e,64,35,4e
363 42 DATA 64,2c,14,0b,ca,7f,ae,f3,85
364 jW DATA 59,69,67,dc,61,73,4c,3d,5d
365 2B DATA 98,00,5e,33,15,09,f5,20,af
366 7H DATA 3c,6c,bc,c5,e2,6b,2e,60,27
367 ol DATA 01,65,6f,09,95,72,39,0d,b3
368 re DATA 26,c1,28,91,07,29,70,ad,ac
369 qd DATA 04,66,f1,f8,9a,e9,68,72,f1
370 VQ DATA 73,8a,40,34,c5,e7,50,e3,66
371 ex DATA 21,e3,ee,60,06,30,5f,b0,00
372 Pe DATA e5,98,73,b9,bd,7f,6c,7c,e3
373 AJ DATA f0,42,1e,09,02,3c,a2,40,b4
374 PD DATA 1f,78,80,07,23,37,08,0c,45
375 rB DATA 20,96,0b,34,cc,3c,b0,30,38
376 uh DATA c7,c3,7c,62,6c,6f,e7,16,f5
377 RK DATA 85,08,20,ab,93,c9,88,36,dc
378 bv DATA 16,e0,00,9e,10,69,c0,9f,70
379 v1 DATA 8d,22,dc,17,b9,53,03,9c,d1
380 13 DATA ae,15,2c,44,34,57,64,32,71
381 Gh DATA ae,04,ce,33,4d,73,91,34,2d
382 y9 DATA 4b,e2,a7,75,5d,7d,24,86,f7
383 QG DATA 10,2b,d9,2d,41,0e,de,11,60
384 12 DATA 72,4a,07,4b,9d,f8,61,20,4b
385 1J DATA e6,6c,6c,9f,3b,91,71,db,07
386 QN DATA c0,23,91,1e,8a,ea,5a,88,03
387 TP DATA 66,9c,4f,52,4b,42,72,b9,43
388 cA DATA 48,71,9f,d0,c0,6d,76,67,db
389 Mt DATA a9,18,76,6f,cd,0a,7a,20,eb
390 Je DATA e4,48,62,79,15,34,72,71,c3
391 Is DATA 89,6e,c9,7b,b2,a7,03,19,20
392 mn DATA 54,2c,d7,c1,5c,5d,5d,45,2a

```

```

393 A1 DATA da,a0,2e,ae,75,4c,49,71,7c
394 5N DATA ac,18,30,4d,eb,a2,98,3a,29
395 v0 DATA d2,6d,6d,65,6f,b8,b1,c1,8d
396 vD DATA 4e,fa,e9,53,44,57,bf,6b,89
397 Lg DATA 6e,2e,24,5a,e4,62,b8,c5,90
398 Y1 DATA 11,e5,4e,6f,5c,09,2b,6e,c2
399 KQ DATA a0,53,3a,5e,16,83,4e,61,6d
400 vA DATA bb,b9,48,1e,6e,04,c6,73,75
401 NJ DATA b9,08,96,83,63,d6,89,53,a0
402 20 DATA ae,2d,53,4c,30,4e,6c,92,66
403 DK DATA 78,18,00,ea,d1,65,b5,85,a4
404 Qc DATA 57,14,64,6f,77,66,95,8b,48
405 Jy DATA 99,38,6f,68,35,e0,2d,3a,73
406 1o DATA 67,72,69,66,66,20,29,21,7a
407 Mh DATA 66,6c,82,28,52,63,71,75,9e
408 3w DATA c1,03,6b,8b,5c,82,6f,3e,1b
409 1g DATA 35,18,38,a8,4e,f4,96,15,b7
410 uh DATA 64,75,ef,b9,65,12,c0,2d,24
411 T4 DATA 53,19,80,ab,86,45,80,11,8e
412 yx DATA 69,8c,3a,c6,72,64,8b,0f,61
413 JQ DATA 79,52,fc,47,96,66,72,16,27
414 vJ DATA cb,97,54,fc,74,7a,b1,bd,4d
415 9T DATA 47,7c,60,55,69,70,8c,91,4c
416 vY DATA c5,86,46,4c,cc,c5,4e,9b,b0
417 3R DATA 6c,9e,4c,2d,66,8b,3d,e4,0e
418 89 DATA ba,bf,21,c9,71,3d,8e,60,03
419 oY DATA 74,2c,9e,49,35,29,58,e2,3c
420 q7 DATA eb,e7,e3,ca,45,8e,0b,8f,62
421 1m DATA 6f,67,5d,94,2b,8f,64,75,72
422 se DATA 8d,94,8b,c9,66,8d,dc,48,78
423 1J DATA 26,04,6e,7a,8c,72,20,31,30
424 o4 DATA 30,20,25,7a,06,c1,e7,06,58
425 54 DATA 4e,2d,45,6e,74,66,70,1e,43
426 v4 DATA 6e,2d,8c,74,68,6f,a0,9a,90
427 fr DATA 60,69,3a,0a,41,4d,49,47,41
428 EV DATA f1,7d,7f,66,57,20,b7,21,41
429 ag DATA 6d,69,67,78,ad,61,6e,d8,55
430 nP DATA 34,11,f0,64,e9,65,6a,83,6b
431 Hp DATA 00,25,01,1d,af,0e,88,1e,3c
432 8k DATA c1,9d,49,e0,8c,69,c1,13,d9
433 1x DATA 19,e2,ca,6b,53,75,6d,5e,ae
434 TK DATA 6b,fa,c0,33,33,6d,61,5c,9d
435 MQ DATA ed,ea,2d,56,65,6b,74,6f,72
436 P5 DATA 20,76,e6,1f,e4,6e,60,63,81
437 Ka DATA 2d,3e,21,63,69,72,73,f4,3f
438 1N DATA 0a,9b,33,33,6d,4d,ef,78,27
439 1e DATA 20,bc,6f,5c,15,80,62,61,72
440 8e DATA 2c,20,64,61,f7,bf,6f,72,65
441 rU DATA 69,74,33,79,08,19,9d,dc,24
442 b1 DATA d9,6a,6c,69,65,72,74,20,21
443 1E DATA 0a,db,fa,5e,b6,4e,03,5b,3e
444 4h DATA 34,04,09,22,7a,23,a0,bb,3d
445 fb DATA cd,97,2a,ac,ed,33,d3,99,65
446 nY DATA 32,50,f4,39,8b,4a,35,18,c0
447 22 DATA 03,e2,00,00,98,42,58,34,3f
448 vI DATA 52,0f,87,6e,45,92,d7,f6,20
449 Yb DATA d2,26,da,76,17,19,1c,22,67
450 vU DATA 6b,05,16,31,1c,8d,7b,81,77
451 0V DATA 7a,13,43,3f,60,06,ba,30,cb
452 R3 DATA fd,22,f2,38,f7,38,d3,83,be
453 Nh DATA fd,51,68,2d,98,74,72,61,63
454 wM DATA 66,24,12,1f,6a,6b,64,69,73
455 21 DATA 66,10,bd,0b,6b,2e,64,65,66
456 31 DATA 14,6a,7c,2c,0b,03,61,c9,d0
457 9p DATA 0a,b8,2d,b4,36,69,29,06,6e
458 vD DATA f0,1e,1e,a0,b8,3b,08,3c,6b
459 CF DATA 1f,b0,ae,fb,07,e7,a2,ee,c1
460 1U DATA 01,96,bc,43,e9,42,13,f8,82
461 S6 DATA 79,4a,13,1d,05,86,16,cb,83
462 eC DATA 3e,d2,9a,4d,3f,0e,43,0b,d3
463 Q2 DATA 0c,92,6f,0a,82,04,20,92,26
464 LY DATA 92,58,88,58,8b,3b,7f,a9,de
465 3b DATA 0c,66,c3,eb,3a,14,43,91,22
466 Tn DATA 80,45,e8,ff,a4,26,8a,0c,79
467 dD DATA a6,7f,42,52,66,16,b5,15,7e
468 41 DATA 9a,83,97,43,1a,43,06,23,09
469 gU DATA 20,0a,58,25,1f,c2,66,0e,ae
470 OM DATA 6a,1d,92,ac,ba,31,10,4b,43
471 vU DATA 1a,0c,96,b7,d6,75,40,6d,28
472 JV DATA 6a,48,86,48,c6,d1,c6,54
473 Yw DATA 3f,0f,d6,18,80,3c,58,04,c5
474 aZ DATA 52,10,7a,3d,47,04,44,96,9b
475 FZ DATA 60,30,25,a4,67,fb,b2,22,13
476 Qu DATA 10,10,42,10,4a,eb,93,58,60

```

```

477 UO DATA bd,3a,8c,05,19,75,21,8c,cb
478 VS DATA 66,37,03,88,ce,4b,37,4d,36
479 9S DATA bb,98,d6,8d,b1,35,15,c6,66
480 vs DATA 0c,66,7e,45,a3,6e,18,b0,05
481 pI DATA 1c,58,02,82,00,d0,ac,a8,2d
482 Je DATA 04,6a,5b,76,05,08,20,4d,90
483 Jw DATA fb,da,6c,87,01,f9,99,66,2e
484 xD DATA b9,ab,f7,66,26,59,36,ad,3e
485 0F DATA b3,50,98,59,cd,67,66,d6,76
486 JS DATA 98,8a,cf,66,0e,0c,98,ec,f8
487 aE DATA 32,66,06,0c,37,c9,6c,ef,ba
488 3w DATA a3,f0,3f,00,0a,e9,e5,59,65
489 Fz DATA dc,c2,20,6f,00,3e,b1,d4,1e
490 Ae DATA 40,9e,65,ac,06,b4,0c,50,dd
491 27 DATA 13,2b,ac,17,de,20,14,b0,bm
492 5L DATA 7c,98,37,a1,a0,f0,99,43,95
493 BB DATA a7,67,73,14,3b,6b,c0,ae,cb
494 bb DATA 23,90,96,4d,2a,5c,09,2b,06
495 e1 DATA e3,02,aa,82,02,82,e5,6e,00
496 0C DATA ef,ae,96,2d,ac,04,5b,97,2b
497 3C DATA 30,cb,9c,2a,de,63,0c,2b,02
498 Nd DATA f1,1f,5a,ac,ea,30,0f,00,ae
499 7v DATA 4d,1c,d4,09,1d,ee,69,0a,ed
500 11 DATA c7,b1,0a,a6,c3,1d,b2,32,22
501 7e DATA 1c,83,28,24,48,d8,1e,be
502 fF DATA a1,1c,04,1a,4c,c4,03,f0,1b
503 2M DATA 36,8c,86,09,cc,44,d1,cc,2c
504 8k DATA 03,f2,5c,33,d5,3a,5a,4f,1b
505 5I DATA 48,64,1a,0a,99,05,b7,1e,5a
506 MM DATA e4,20,f0,e5,35,01,b2,ba,cd
507 6C DATA 60,b6,cc,01,4e,af,30,8f,40
508 IG DATA e4,73,2b,1f,c1,22,8b,87,25
509 4k DATA df,d6,0a,cd,e3,1a,a6,2c,1e
510 vu DATA ec,c2,64,39,01,ce,33,24,03
511 30 DATA 6c,9c,87,90,6a,87,1a,8e,09
512 BS DATA 07,c6,1a,bc,f9,3c,9b,56,03
513 0T DATA 9a,09,d2,99,63,c7,19,f4,1a
514 sv DATA 50,27,cf,c7,66,ea,c0,03,f0
515 tc DATA 19,1e,dc,6e,6c,15,76,02,de
516 81 DATA 94,5c,ea,bc,42,32,83,8c,44
517 kh DATA a5,9f,60,f3,10,2a,bc,07,c2
518 BC DATA 54,9e,c7,05,f4,16,0c,ae,36
519 wa DATA 09,ef,ff,fc,67,c4,4a,fb,b4
520 3U DATA 14,66,be,ba,43,14,6f,ad,62
521 BP DATA 66,18,88,64,1e,ee,60,ae,c5
522 br DATA b0,19,e2,c2,ff,2c,09,87,72
523 eL DATA 23,15,e0,28,c5,e2,02,12,52
524 JB DATA fd,0e,2b,0d,ce,d6,50,38,80
525 55 DATA 06,61,0a,66,77,05,61,b1,01
526 1P DATA f4,60,9d,10,00,f0,0b,d5,30
527 2W DATA dc,b1,8c,45,5a,16,a0,47,60
528 gI DATA 0e,68,49,14,64,20,be,91,2b
529 DS DATA be,02,14,a5,1c,60,7a,e2,e7
530 nq DATA ee,1e,30,d1,ad,06,3a,60,59
531 4Y DATA 06,9f,e2,77,10,0f,ae,65,e8
532 JH DATA 14,4a,39,cf,e3,5e,22,f2,0c
533 5e DATA 01,9a,f9,0f,ce,71,fc,0a,87
534 fg DATA 44,24,08,76,01,e3,04,74,01
535 33 DATA 9e,19,1e,01,6b,62,e9,9e,9a
536 21 DATA 53,08,6c,cc,8d,bd,0b,0b,94
537 q1 DATA e0,91,08,b3,35,0e,00,98,8d
538 wa DATA 66,ac,39,02,80,76,64,fb,68
539 oS DATA e4,2f,2c,5f,f0,18,ee,20,1f
540 1d DATA ba,2e,66,2f,24,5f,7f,1d,06
541 7r DATA 20,08,20,5f,7b,1c,98,21,dc
542 1L DATA 3a,78,d8,81,23,c0,67,72,18
543 7x DATA 83,f7,56,55,7b,98,82,27,18
544 3c DATA 68,c2,12,b1,ee,8a,00,06,24
545 7p DATA 2f,29,6d,1a,18,3a,0c,6b,21
546 2N DATA 55,67,16,99,12,ee,66,0a,df
547 25 DATA 86,97,ad,0c,09,80,0c,5b,75
548 71 DATA e0,08,47,38,c5,04,36,73,c0
549 52 DATA c6,d1,c0,2a,48,23,48,c0,df
550 31 DATA d5,9e,ac,13,1a,4c,60,56,51
551 9q DATA ea,eb,08,9c,33,92,70,d9,ad
552 1t DATA 28,23,5f,00,24,7c,06,69,15
553 om DATA 08,28,15,43,1f,c3,40,67,93,6a
554 0L DATA fe,3e,66,ee,5d,18,3b,03,2a
555 3M DATA 02,36,ff,7e,71,d7,2c,62,1c
556 xz DATA 29,a1,06,ff,56,44,d0,98,64
557 ac DATA 02,52,80,f5,3b,b0,f8,46,b0

```

Listing. »VirusControl« macht Schluss mit Viren aller Art (Fortsetzung)



558 2I DATA 46,06,42,a8,7c,1f,e4,2b,f6
559 NC DATA 7e,fb,2d,b2,75,99,7c,b8,30,3c
560 Qh DATA 03,ff,ee,e9,0c,94,52,41,57
561 NE DATA 3a,66,10,c3,3f,20,65,06,b4
562 YI DATA e7,62,7f,65,59,41,bc,00,2e
563 BQ DATA 52,88,fc,cd,e4,ac,7d,76,ce
564 An DATA 5c,67,0c,75,10,06,ea,29,00
565 ZH DATA 30,31,1f,81,69,32,26,36,94
566 T6 DATA a7,6a,ed,b1,c4,15,1d,98,32
567 wG DATA 0b,20,f1,99,83,21,b0,ce,42
568 Mx DATA 22,04,66,21,29,40,33,81,8d
569 vM DATA 9c,10,80,d9,76,61,00,17,34
570 YX DATA 71,6a,ce,2a,cb,39,60,dd,0f
571 Qe DATA 19,63,07,0c,15,07,44,4f,33
572 Ay DATA 00,54,8a,16,b8,8f,bf,f2,70
573 Zf DATA d6,f6,ba,d1,06,28,e0,39,25
574 Pn DATA 67,85,36,d3,02,ae,58,92,ce
575 Or DATA 31,d7,2c,cf,6a,4d,75,0e,99
576 yG DATA d7,da,25,a7,99,10,87,4a,e3
577 go DATA 81,66,f8,66,0d,3a,0c,00,a0
578 lb DATA 23,c8,b9,03,04,32,3c,00,28
579 jG DATA 14,9a,0f,b4,b6,22,83,05,67
580 pa DATA f8,51,c9,ff,0f,e1,67,60,d0
581 BQ DATA 13,ae,fd,d8,2c,40,f2,14,00
582 by DATA 46,23,ee,9b,47,67,80,42,28
583 jW DATA 07,88,88,33,fc,80,81,3b,df
584 Ot DATA f0,96,22,c8,65,24,8d,fe,62
585 Ym DATA 6e,b6,aa,43,fa,00,18,75,cf
586 Ec DATA 80,a0,b5,03,0a,20,96,e9,6a
587 rC DATA 16,70,66,93,75,70,ff,60,fa
588 xC DATA 64,6f,f9,47,00,67,de,70,68
589 F4 DATA 04,5e,73,2e,11,50,62,72,61
590 Y1 DATA 72,79,88,ff,83,80,00,0f,ed
591 m1 DATA 9b,5e,da,93,ce,80,34,77,49
592 MJ DATA 09,77,6f,0c,6c,69,6e,6b,18
593 T1 DATA 47,4d,30,43,61,73,00,7f,20
594 jQ DATA 6e,69,fc,74,20,09,c3,64,72
595 2u DATA 75,65,63,6b,74,f8,16,77,69
596 jX DATA 72,64,20,62,ea,71,42,6f,6f
597 xW DATA 70,7f,b8,61,75,74,6f,6d,61
598 qV DATA 74,69,c1,df,20,46,41,53,54
599 jW DATA 4d,45,4d,14,ff,e7,62,67,65
600 FN DATA 73,63,68,61,6c,24,74,65,74,66
601 zD DATA 47,36,49,52,aa,29,2d,43,4f
602 Bb DATA 4e,54,52,fa,1a,56,31,2e,33
603 JA DATA 20,76,6f,6e,20,49,fc,d4,3b
604 j1 DATA 20,4e,49,50,50,47,45,4e,e3
605 gT DATA 7f,36,36,35,37,20,47,65,72
606 Sg DATA 73,68,65,69,6d,20,38,2c,20
607 qM DATA 30,36,38,34,33,2f,31,35,32
608 tQ DATA 32,00,0c,70,a0,3f,39,00,bf
609 Xx DATA ec,01,67,54,80,cf,7e,20,41
610 Am DATA 0c,10,00,2a,73,cf,7e,67,44
611 tw DATA 33,4b,74,fe,99,e7,66,fa,92
612 a9 DATA 07,f0,ce,22,b5,32,7a,1a,50
613 vF DATA cd,a1,eb,30,69,19,e3,52,10
614 bg DATA da,41,11,0a,53,fa,f5,56,5e
615 fg DATA dd,20,00,fd,ac,67,02,d4,04
616 yF DATA c8,06,00,28,23,96,78,67,f6
617 Ox DATA b4,19,1a,12,20,fa,5b,14,8c
618 2W DATA 97,0b,ee,69,09,9e,2d,0c,a1
619 At DATA b9,1a,b9,ec,03,d2,b2,2e,ce
620 3r DATA 5b,16,30,2a,70,e4,20,16,1f
621 Or DATA 26,d4,ba,58,79,f1,ee,2e,b9
622 91 DATA f6,62,58,2e,ae,82,87,1a,20
623 zN DATA 8a,d1,ed,70,31,60,42,90,77
624 hS DATA 28,31,0b,93,48,2c,7a,19,b4
625 qp DATA f8,a3,28,b1,ce,e8,67,3e,d0
626 Aw DATA 24,ff,7c,30,5d,4e,49,6b,d8
627 xH DATA ee,ff,ae,21,4c,80,1e,21,6e
628 ee DATA 81,35,c8,dd,c2,a7,99,94,88
629 PU DATA 75,60,fe,f8,3d,7c,ae,f9,ff
630 xQ DATA e2,2d,48,ff,e4,cd,4c,a1,5f
631 wp DATA 0b,67,24,32,6b,ff,6a,53,f9
632 uP DATA b1,16,b6,c9,03,5a,c5,18,f4
633 GO DATA eb,b0,3c,00,9a,42,37,d5,2e
634 Xs DATA ec,c1,3a,fc,a4,22,4e,20,bb
635 9P DATA 3f,f4,e5,22,0f,5c,14,4e,c6
636 eb1 DATA 16,bd,18,cb,24,17,ea,95,18
637 zY DATA c9,51,e3,6e,c2,fd,8d,05,d1
638 GS DATA e4,09,e2,a3,cf,f6,54,47,16

639 6V DATA b6,ec,f3,eb,42,6c,cc,f3,03
640 -5 DATA f0,19,1a,2c,60,7d,7e,9d,42
641 FW DATA 50,02,5f,1c,15,e8,c7,18,5a
642 EG DATA f9,e7,cf,0a,f5,f9,42,02,75
643 yX DATA 06,fa,42,2a,71,16,fa,60,59
644 Q4 DATA 2c,90,f9,e8,b7,d9,5f,1a,18
645 UK DATA 53,4a,98,66,08,eb,95,6a,42
646 xa DATA 12,60,10,4a,12,66,0c,14,ee
647 zJ DATA df,49,8b,23,c7,92,02,3c,20
648 -c DATA d9,56,6a,b0,f6,cf,fd,18,62
649 dt DATA 91,37,5e,26,90,b5,d0,67,26
650 Bw DATA ea,6f,96,d0,27,65,67,69,df
651 bQ DATA b6,6e,49,4f,e9,55,02,04,60
652 9q DATA 0a,bb,2f,c7,3c,da,30,50,5e
653 zC DATA 60,ae,9f,5f,0a,22,5a,18,2c
654 Pt DATA 18,86,18,28,e1,04,98,b0,0a
655 JP DATA 4d,65,64,44,46,00,a0,60,fc
656 24 DATA 26,40,1b,26,b4,0f,74,d1,17
657 bS DATA fa,cb,da,23,26,74,56,a0,4b
658 rD DATA 67,30,1f,93,74,72,30,10,15
659 sz DATA 2e,66,18,04,1c,02,b0,0e,d0
660 iZ DATA d8,07,0c,9a,60,f4,97,4b,69
661 gG DATA 63,6b,a0,13,68,65,63,d8,24
662 HM DATA 6b,73,75,9d,99,14,fc,ae,db
663 eb DATA b5,be,67,2c,da,2a,40,0b,ae
664 5G DATA 68,04,9a,5d,b0,c2,0b,81,4c
665 jP DATA 44,d9,00,01,01,30,cf,49,5e
666 wI DATA 1d,f4,2e,75,28,24,7e,17,72
667 qI DATA 29,4f,17,6e,26,88,b5,c8,67
668 Ua DATA 1a,2e,fc,31,8e,94,43,4f,4f
669 zM DATA a0,3d,fa,e5,90,20,bc,19,20
670 PY DATA c0,9c,5a,e7,c2,2c,30,70,e4
671 rY DATA e3,56,41,32,81,62,12,68,9c
672 3Z DATA 0c,5d,80,6a,84,9c,fd,d4,66
673 pE DATA ae,60,30,7b,12,4c,41,09,b6
674 tE DATA 45,20,82,01,5c,b7,f6,08,00
675 9L DATA 67,06,f1,0b,22,3a,88,25,2e
676 wJ DATA b2,38,ae,4b,f2,18,31,ea,b2
677 uJ DATA 18,38,09,00,aa,96,1b,be,41
678 09 DATA c8,23,7e,72,30,52,81,7b,07
679 QA DATA 06,95,ae,60,2a,99,47,18,b4
680 Ym DATA 1f,18,76,4b,e6,ae,81,21,f6
681 mb DATA f8,06,15,41,00,0a,29,0f,20
682 hs DATA da,20,92,e0,c7,bd,68,da,32
683 zB DATA c0,f0,a1,af,04,72,00,5f,9e
684 dD DATA 9a,0e,50,88,06,42,11,e5,93,2e
685 gd DATA 1a,24,f2,60,2e,db,01,76,3c
686 oh DATA 26,52,24,a0,95,f7,f8,35,3c
687 uC DATA b7,88,67,de,f3,82,33,0e,62
688 fd DATA 49,79,4c,a0,50,20,98,5d
689 q1 DATA 0f,b0,b8,2a,61,1a,6b,37,16
690 xX DATA 4a,11,ed,10,ae,59,09,1d,7a
691 Ty DATA 61,0e,ba,46,06,42,11,b7,67
692 UY DATA 64,9e,c5,21,66,60,03,21,ae
693 Q3 DATA 47,fa,1a,f1,93,b8,f7,b8,70
694 89 DATA 08,2b,ed,c5,5f,4e,75,18,3f
695 ku DATA 20,87,ad,cf,b9,00,20,67,ee
696 U9 DATA e4,ee,fa,38,f3,b6,fd,05,78
697 Mx DATA 4d,72,34,5b,ee,f0,45,6a,05
698 qG DATA ce,4a,06,d4,c9,76,6f,98,c3
699 SZ DATA 1d,23,b8,7b,c0,95,ce,4a,e0
700 jP DATA fa,55,c9,0c,0c,7c,0e,38,ab
701 eQ DATA 8b,29,3c,23,49,00,12,28,69
702 OE DATA 00,22,28,54,f8,7b,f8,dc,70
703 pq DATA be,8a,ac,3b,70,34,6a,04,62
704 It DATA 75,80,93,e2,71,47,46,66,04
705 To DATA 2e,00,66,12,fd,1a,58,d9,b5
706 R4 DATA 14,50,20,7d,0e,f7,3e,cb,51
707 f3 DATA b2,4a,6e,d9,28,ca,14,35,ed
708 mQ DATA 3c,56,70,3d,18,32,18,b8,23
709 vt DATA 08,58,a0,00,58,8b,32,76,81
710 Ch DATA 3a,b2,f5,d3,c0,0c,1f,71,2a
711 HU DATA 72,37,37,56,0f,f4,eb,08,05
712 HO DATA 00,1f,67,12,bd,95,72,45,99
713 to DATA 5a,18,5e,e4,ba,84,28,db,dc
714 2a DATA 3e,80,ff,2f,0e,6f,19,2c,c8
715 vL DATA 00,6f,76,6e,36,04,20,40,78
716 Q3 DATA 16,91,15,1c,3a,28,00,7a,4b
717 7V DATA 9c,b3,81,1b,09,67,c4,da,59
718 Xh DATA 13,6f,16,3b,1a,c0,0c,ae,b0
719 M1 DATA 41,cf,60,ae,c0,51,14,66,10
720 sw DATA 2c,57,e4,21,e0,9e,ba,82,85
721 dD DATA 2d,26,f6,7e,56,5d,e6,82,ec
722 Rr DATA 86,8d,2d,bd,fd,25,81,fe,86

723 4r DATA b9,44,da,2c,5f,20,47,f3,f8
724 iz DATA 32,14,38,ce,4a,24,47,20,aa
725 xM DATA dc,67,65,d2,18,72,04,a2,06
726 Kt DATA d1,c1,d6,37,e1,71,9d,b8,72
727 7B DATA b5,e1,0e,d6,dc,0a,42,b3,28
728 Jo DATA 83,10,0a,f1,2c,8a,f7,96,96
729 8z DATA 53,e8,0c,18,e6,64,4a,58,5d
730 VB DATA 2e,27,9d,8a,13,c4,b3,dd,ed
731 0L DATA 22,90,0c,78,eb,c8,6a,66,76
732 FX DATA eb,86,60,be,15,93,59,2a,e0
733 rP DATA 7d,ea,74,9e,49,fa,14,e4,ed
734 xR DATA 57,e6,bd,02,66,14,da,d1,00
735 Zq DATA a2,da,ae,b4,b6,c0,d0,29,6a
736 aZ DATA a7,00,70,65,48,18,80,08,ae
737 xm DATA e7,e2,98,16,fa,d1,22,90,0a
738 Ho DATA 22,c3,0e,12,0e,25,03,f0,0c
739 kv DATA e2,bc,4d,dd,73,33,47,a0,09
740 jh DATA 0a,73,d9,1e,ad,f6,b4,96,ef
741 tN DATA be,f0,66,8f,ff,45,8a,f3,f4
742 Eg DATA 75,66,8a,f3,f2,99,0f,04,66
743 sO DATA 24,d8,03,92,72,51,fd,dc,b9
744 x7 DATA 0e,35,c0,f6,ae,70,31,d6,a9
745 rQ DATA 76,1a,f5,c8,04,eb,fd,10,f0
746 An DATA 05,57,09,84,8b,08,68,f3,07
747 zG DATA 01,cc,2c,ad,f6,b4,96,ef,fa
748 zN DATA a0,60,d0,ee,a7,06,26,aa,77
749 qJ DATA 9f,dc,fa,07,66,94,8b,35,f3
750 rJ DATA b6,06,4e,b0,ca,f5,08,66,1c
751 qC DATA 6b,b0,c3,8c,4a,fa,c0,ae,52
752 co DATA e3,12,7a,ce,59,e3,0e,92,ac
753 cJ DATA f6,26,21,7e,0b,e4,19,b0,3a
754 -a DATA e3,86,09,12,ba,67,4d,0d,18
755 pR DATA ce,04,33,22,e2,49,01,e4,b0
756 Ao DATA 71,46,f6,78,29,f3,0a,c4,45
757 Bw DATA b5,4d,b2,32,c8,ad,f8,69,f0
758 Yk DATA 00,0f,13,00,2f,b1,01,6a,96
759 FF DATA 4d,31,51,00,09,56,3c,2b,66
760 G1 DATA c8,8a,93,fa,69,01,da,35,f2
761 mZ DATA 71,44,09,42,96,61,65,fd,ce
762 gQ DATA 40,c0,24,0c,58,c3,61,66,95
763 vC DATA 83,75,2d,0c,54,50,f2,de,4c
764 kb DATA 9f,e3,11,fc,42,90,7e,1d,5a
765 jY DATA ab,80,12,70,78,0d,ce,05,0b
766 Qs DATA 26,91,37,61,bd,ea,6d,19,c0
767 nH DATA 1a,4b,06,e3,f6,e0,02,5e,34
768 19 DATA 07,ac,f5,04,f2,4c,ac,37,27
769 OH DATA b8,d4,23,35,fa,24,d8,66,7f
770 uT DATA d5,0c,e9,fa,e4,0a,76,46,db
771 VP DATA d9,80,55,43,16,de,c0,0a,b8
772 vs DATA 32,d7,ed,24,4d,0c,90,4a,84
773 w1 DATA 66,15,aa,8a,b5,88,67,d9,af
774 xk DATA 68,d9,6e,52,8a,93,65,aa,79
775 aZ DATA e7,c0,d6,68,9e,8a,8a,91,03
776 R6 DATA 00,88,63,6e,da,fd,0a,2a,b7
777 Po DATA f1,d2,38,96,a9,44,33,d3,81
778 B3 DATA 32,17,62,14,29,17,0c,66,4e
779 tz DATA ea,87,07,02,96,2d,e6,d9,42
780 8Z DATA 60,9c,50,0a,2c,2a,bc,46,52
781 Tt DATA 45,45,60,16,20,4c,eb,7f,8f
782 EM DATA 2e,82,11,02,f0,36,57,d2,08
783 LX DATA 28,24,0c,76,20,db,6b,05,60
784 W7 DATA 10,24,3a,10,a0,67,0a,04,82
785 qS DATA 78,fd,48,1d,03,06,24,0d,db
786 tA DATA c8,3c,4e,50,16,e0,71,c2,f2
787 r6 DATA ba,dd,b3,20,f1,42,f8,0d,66
788 rZ DATA 0c,69,ef,cf,bc,8a,fa,50,c3
789 ax DATA 9a,fa,7c,e2,ef,0e,59,19,04
790 xK DATA fa,58,7b,c6,10,58,19,50,fa
791 jV DATA 80,66,46,87,59,2e,ed,12,88
792 2O DATA a8,b3,20,0c,98,03,28,40,b3
793 BD DATA 81,ce,3c,76,0a,cc,66,88,18
794 31 DATA 11,42,e7,67,ed,ae,33,1c,97
795 Rv DATA 42,d4,22,53,56,93,6a,79,17
796 xY DATA 06,0a,2b,8f,8d,c3,a5,2b,9a
797 35 DATA 68,75,40,3e,7b,aa,80,47,38
798 qQ DATA a1,9a,74,34,94,ac,75,59,72
799 6W DATA 01,16,94,1d,70,0b,2c,01,1e
800 -e DATA 1e,8c,53,84,73,83,71,87,72
801 TJ DATA 6b,70,c3,1b,93,98,06,3d,ea
802 LF DATA 00,ae,2f,02,06,a5,07,61,01
803 jk DATA 04,24,40,5d,ca,u2,9a,b9,8f
804 -D DATA 70,8b,07,ce,22,61,27,08,ce
805 zh DATA 39,78,94,77,1b,cf,0a,20,42
806 2y DATA 41,8e,f9,00,4d,b3,b6,51,eb


```

807 ip DATA ff,de,3c,d3,5c,db,e4,2f,0a
808 iu DATA 7b,ff,ed,0f,77,95,51,72,71
809 iu DATA 69,2e,28,06,97,6c,9f,ed,20
810 bp DATA 28,57,20,57,f2,e3,b0,1a,96
811 rj DATA c7,ec,b9,0b,72,e0,b1,62,60
812 jo DATA d0,de,c0,32,0b,b7,e5,fb,04
813 54 DATA 58,16,85,11,78,e5,82,d2,4f
814 vq DATA b8,35,c2,2d,94,e6,f2,c1,0c
815 pn DATA e4,99,42,d6,f1,1a,99,42,d6
816 g3 DATA 25,7e,35,87,6a,04,5e,c2,2e
817 ct DATA ef,72,22,ad,4d,27,ab,4a,0b
818 k0 DATA ec,0c,55,e3,22,0f,1e,30,09
819 g8 DATA e8,10,7b,9c,40,db,b5,10,5e
820 pg DATA 2c,0d,02,c2,65,c8,0c,95,f2
821 4b DATA 86,f3,67,23,7b,69,59,95,18
822 t9 DATA f7,e3,6d,03,c4,68,00,c6,0d
823 d9 DATA cc,a0,3b,22,da,26,47,2a,14
824 j8 DATA 33,43,58,0f,f3,67,07,e0,03
825 lj DATA de,a6,10,0c,ad,5c,46,82,0b
826 yl DATA 8e,71,56,bd,01,0e,99,69,40
827 w7 DATA d3,6d,e1,01,de,2c,27,74,02
828 q8 DATA b7,38,1a,ea,29,23,b2,ca,72
829 p9 DATA c2,10,4d,24,b2,21,86,02,3c
830 ei DATA 73,20,c0,57,32,c4,81,13,24
831 q6 DATA 00,b9,a7,60,14,10,80,fb,04
832 lv DATA 38,26,2f,b6,d2,96,82,c4,4a
833 w7 DATA 83,06,0b,f0,26,57,d7,c3,24
834 q1 DATA 57,fd,0c,c0,20,bc,e9,1e,7a
835 2f DATA 78,20,3c,91,76,df,20,fb,4f
836 ls DATA de,40,59,8a,d4,83,03,e8,66
837 iQ DATA 1c,f6,e3,32,32,b8,e9,19,be
838 zj DATA 07,82,78,24,0c,9a,69,a3,d9
839 De DATA 6c,ed,da,d6,27,16,d0,aa,ff
840 qj DATA fe,5b,be,09,02,5e,80,c0,d0
841 Jo DATA d1,90,4a,90,66,02,56,90,60
842 Sc DATA ba,16,80,ff,45,2a,0d,80,08
843 qu DATA b5,55,8a,45,52,9d,4d,a1,d5
844 xu DATA 08,b4,88,66,44,60,88,ff,c4
845 za DATA 03,da,66,3c,7a,80,b0,d0,02
846 Qv DATA aa,2a,20,49,22,5f,20,1f,e5
847 Fc DATA 3e,dd,0e,e0,2e,1d,d1,48,5b
848 KQ DATA 42,0f,99,38,97,50,bd,de,14
849 Q0 DATA 01,52,c4,ee,58,83,c4,21,40
850 RS DATA 6d,8f,e5,7a,0c,fe,76,e5,de
851 QK DATA 18,aa,33,e0,c0,8c,d9,6f,67
852 e3 DATA a4,c3,81,15,6d,99,65,32,57
853 o0 DATA 15,4a,09,8e,3e,aa,06,03,dc
854 j9 DATA 37,ff,8a,58,56,0f,71,16,0b
855 mq DATA 26,96,9d,e1,ff,7a,dc,08,d5
856 XP DATA 0b,ba,18,cd,ac,24,b5,ed,de
857 4a DATA ee,1c,f2,9d,ae,a3,2f,fa,b5
858 1d DATA ac,02,aa,66,2e,71,2f,82,48
859 o0 DATA 20,3a,0c,7a,b0,92,66,d4,ed
860 av DATA ff,27,a4,37,8a,ba,eb,96,23
861 Q0 DATA 55,0d,a2,ff,26,e2,27,90,fa
862 ee DATA 78,1e,74,cb,37,b7,03,d2,22
863 7w DATA 49,15,e9,76,ee,32,b9,d1,e5
864 OP DATA e1,ed,8b,00,af,82,5a,ca,c8
865 54 DATA 43,de,0c,2a,7c,65,03,ee,0b
866 HO DATA 1d,84,7b,b3,c4,66,2a,5a,f6
867 2H DATA a7,14,ea,75,3c,05,f0,7a,14
868 12 DATA c4,96,1f,64,e8,97,91,0c,22
869 oW DATA 10,f4,8b,ae,b5,ce,58,07,ae
870 ob DATA 22,00,24,17,26,d8,df,d4,4d
871 uJ DATA 1d,27,e0,0e,24,59,0a,32,85
872 Sr DATA b5,fe,86,4a,84,67,1a,78,ff
873 Kv DATA e8,40,17,e4,32,f1,01,9a,fe
874 5w DATA 5c,09,23,09,eb,30,f8,8c,5b
875 Qj DATA de,cd,4b,1f,e5,4b,ff,4f,01
876 Wy DATA c9,b6,10,10,42,10,4a,00,66
877 m1 DATA 8a,5f,38,40,ec,2e,16,5e,0f
878 nS DATA 66,28,60,23,c0,0b,29,39,41
879 b1 DATA ee,bb,44,4a,82,c0,68,6a,f7
880 NO DATA 6e,23,ee,99,58,f3,ba,0e,56
881 o6 DATA 62,b5,10,aa,cf,fd,d2,69,98
882 dH DATA 6b,f7,36,e1,55,15,7d,ee,e1
883 EB DATA 4e,58,db,aa,b8,56,16,66,e5
884 9a DATA 34,91,e2,c8,5a,10,17,18,e1
885 aa DATA 00,a0,0c,45,00,17,66,1e,fa
886 P3 DATA bc,34,0a,b2,46,de,4d,b8,86
887 UQ DATA 12,9f,87,08,74,5a,06,16,ce
888 K6 DATA e1,09,7c,58,e1,f4,ce,c3,38
889 Bb DATA 0a,2f,28,7a,ee,9e,55,33,e1
890 o8 DATA fc,cb,53,e4,60,30,6c,a3,ee

```

```

891 Yw DATA 97,22,60,22,7d,0c,e0,dc,7d
892 1u DATA a4,60,10,da,00,74,21,ca,94
893 od DATA 13,e6,85,41,14,22,58,40,14
894 jJ DATA be,0c,ec,16,b2,07,0f,5c,0e
895 yV DATA f0,ba,84,12,4e,74,4e,ba,e4
896 ES DATA 14,12,32,60,d0,de,2d,e0,86
897 Is DATA 55,0e,3e,e1,a1,c2,39,3d,0e
898 YD DATA be,c4,1a,eb,1a,d9,8c,27,00
899 GY DATA 4a,66,be,91,b5,2c,e0,5a,bd
900 gr DATA 12,ea,48,60,d2,d5,5e,16,f0
901 o0 DATA 40,5e,52,f5,aa,60,b8,6a,4b
902 PC DATA 05,78,34,ed,26,7c,38,4d,97
903 Be DATA 14,29,0f,8d,92,9e,32,41,01
904 qN DATA d6,60,1a,df,b5,06,78,38,a2
905 kW DATA d8,50,12,59,b3,86,fa,fe,43
906 v0 DATA 29,13,60,c6,ff,b4,68,1e,29
907 q3 DATA e0,9e,93,13,64,60,e9,32,5a
908 YD DATA 1f,06,2a,60,78,4e,a7,65,64
909 DU DATA 67,83,2d,33,17,ba,36,fb,b1
910 GD DATA eb,92,11,46,e5,22,b6,3e,2a
911 QA DATA 41,bf,ed,06,76,e8,60,9c,68
912 3a DATA 3b,10,b8,60,96,e8,8d,04,10
913 dF DATA 5c,3c,47,ea,b5,98,c0,84,b1
914 v1 DATA db,53,33,02,3c,11,20,0e,de
915 RI DATA c2,aa,12,c0,ea,41,bb,26,0f
916 6V DATA 47,e2,5b,4e,5a,16,40,0b,60
917 aE DATA 68,0b,ac,4c,3b,e9,11,87,20
918 w8 DATA f0,42,41,44,20,b9,df,81,f8
919 wq DATA f9,a3,04,56,0f,bc,60,00,fe
920 24 DATA 7c,99,fe,9e,56,9e,dc,ae,ec
921 1A DATA 84,88,20,80,ed,c4,76,8d,aa
922 1P DATA 5b,18,be,b5,9a,75,16,38,28
923 Na DATA ae,98,73,ee,9a,97,eb,de,e6
924 KY DATA 54,18,00,3c,12,5a,32,37,7e
925 WY DATA 46,eb,08,d2,9d,93,c6,8a,3d
926 LG DATA 26,34,6e,ba,ab,06,31,e6,8c
927 SX DATA 42,33,3c,07,f4,71,08,28,33
928 YK DATA 7c,00,09,00,1c,ed,0f,81,15
929 ZD DATA 9c,95,84,c6,31,c4,39,80,98
930 ev DATA 6a,ce,00,3c,f0,58,97,98,41
931 QS DATA 81,62,1c,1c,e0,da,0d,0e,f0
932 TO DATA d0,12,01,78,97,b9,6f,4a,87
933 Y6 DATA 06,20,50,f4,6f,38,29,02,86
934 Ba DATA 05,4f,c2,b9,7d,a2,40,92,71
935 1a DATA 9a,e9,8c,05,07,e5,29,02,66
936 QJ DATA 0a,02,2a,42,ae,00,2a,f4,17
937 ug DATA b0,3a,46,c9,da,9e,b8,d4,9a
938 YR DATA 59,5f,65,08,20,7a,07,a4,20
939 OI DATA ba,07,ae,f5,f3,00,f0,80,07
940 Mb DATA 7a,e8,00,92,45,f8,fb,70,06
941 oe DATA 80,bf,08,b9,24,64,e6,80,04
942 Pq DATA 24,0d,8d,a2,81,ea,ea,23,98
943 ZE DATA 47,fa,23,d4,45,f8,38,58,70
944 w6 DATA 0f,0c,90,e2,a3,c6,64,40,88
945 ax DATA 03,91,58,8b,60,0a,24,90,26
946 DK DATA d8,d5,78,7e,6a,99,0c,8d,e0
947 fs DATA a2,0a,0f,00,33,fe,c0,e2,2f
948 mL DATA df,00,9a,7b,3a,a8,38,96,3a
949 ca DATA 2a,28,90,1d,20,bc,10,b7,01
950 90 DATA 53,14,1a,e3,cc,20,08,77,be
951 dg DATA 20,b9,45,fe,38,cb,b3,b0,5e
952 ma DATA 12,1c,0a,88,83,1c,d1,ed,61
953 Q1 DATA b2,01,60,42,90,2e,8b,2d,7a
954 DC DATA 06,5b,2e,41,ee,e0,59,42,40
955 ht DATA 72,17,d0,58,d4,e7,fa,46,40
956 UK DATA 30,80,91,57,30,d0,7d,ff,42
957 1Z DATA 98,8d,00,20,4d,24,48,60,ee
958 OF DATA ef,c5,60,96,34,61,4a,2e,8d
959 Gy DATA 1b,03,48,09,be,70,3a,c0,c6
960 ya DATA d5,c0,61,02,e9,47,70,1f,14
961 BY DATA d8,51,c8,ff,fc,c0,af,15,e3
962 G7 DATA 28,41,20,41,4e,0d,33,02,52
963 2D DATA 30,cd,40,b3,80,84,c1,5c,0e
964 Bu DATA 18,36,66,f6,20,6e,00,22,4c
965 k2 DATA 5f,18,d4,61,04,26,4c,de,06
966 yC DATA ec,e3,1f,28,e5,88,24,40,12
967 Bj DATA 1a,48,81,53,41,76,bf,66,12
968 Gg DATA 51,e9,ff,06,4f,13,00,3a,e5
969 IT DATA 56,1e,4c,60,20,70,ff,39,9d
970 Fk DATA 40,59,b1,e4,04,19,40,53,81
971 Ug DATA 69,0a,67,da,06,23,78,bf,5a
972 w7 DATA dc,43,28,e4,70,d5,70,ee,da
973 yC DATA ac,0c,e2,da,c2,66,6e,b5,15
974 oH DATA 0e,92,98,f5,66,b6,f8,76,62

```

```

975 s1 DATA 10,b7,ca,65,b5,88,8b,33,b0
976 va DATA 4c,04,5e,3b,00,14,b0,9a,67
977 1f DATA 26,db,7b,0c,22,75,1e,36,a0
978 FD DATA 2f,22,e2,0e,3e,60,6f,26,ee
979 ge DATA 00,8e,1d,6e,a0,08,20,4c,98
980 Eb DATA 81,0c,8d,c6,df,94,07,e4,44
981 1J DATA 36,04,18,60,0f,4c,4d,fa,16
982 Sk DATA ee,f8,68,c3,e1,b1,48,58,8a
983 Qy DATA 0d,f8,3d,1b,e7,16,ca,60,1e
984 56 DATA b5,89,ee,30,60,12,b8,13,05
985 J2 DATA c4,60,08,71,a7,e1,74,e9,22
986 nK DATA 70,84,37,b1,fd,d8,4c,02,33
987 mN DATA 4c,98,e1,ba,2c,5b,5d,8c,63
988 pr DATA de,22,4e,ee,0a,ee,fe,62,dd
989 B9 DATA 43,11,e5,17,05,0c,d1,c6,42
990 uy DATA 19,3e,35,16,16,9d,37,8c,23
991 k2 DATA 95,a0,d4,59,3f,e5,b6,09,ab
992 Xa DATA 80,35,02,04,dc,52,10,0d,10
993 9a DATA be,7f,66,10,e2,41,5a,04,a4
994 KJ DATA 71,97,ec,4f,22,01,76,10,13
995 Fu DATA 05,28,88,83,6a,1a,70,d4,a0
996 ap DATA 72,32,e8,6d,81,09,bf,80,db
997 Zb DATA 89,0a,c0,75,a2,e3,99,b2,7c
998 Y DATA 0f,cb,a6,ae,25,f0,20,60,ff
999 1V DATA 28,9c,35,03,de,26,49,e5,00
1000 3Q DATA 00,42,9b,38,d1,94,26,ae,61
1001 6k DATA 0c,2b,00,32,02,12,67,08,c4
1002 Pa DATA 3f,c9,60,1b,4d,20,b9,df,81,f8
1003 Zu DATA 96,7d,92,5b,2c,03,8c,2c,97
1004 83 DATA c9,8d,e9,62,12,e1,27,10,cc
1005 Yn DATA 56,32,01,0a,b3,00,7e,d1,ff
1006 1G DATA 00,e0,36,07,15,bc,7b,1a,6a
1007 wH DATA cc,d8,c6,62,21,4a,b0,15,84
1008 16 DATA 0e,00,fe,d0,af,4f,76,e5,8a
1009 p0 DATA 01,78,70,98,20,6d,58,23,10
1010 14 DATA 66,e2,4c,9e,04,3b,80,2a,0b
1011 fh DATA 2f,96,49,fe,a0,20,1f,d6,11
1012 fv DATA 66,06,c4,4c,60,c4,20,6a,6f
1013 yE DATA 40,45,31,a1,0e,b5,c8,9a,00
1014 Jq DATA 66,b4,f3,0a,54,45,04,06,13
1015 1b DATA 07,6e,35,70,a8,41,72,31,59
1016 zb DATA 00,74,79,35,09,90,60,4b,80
1017 Dn DATA 6d,c8,2e,20,40,4a,80,f9,a2
1018 ac DATA 78,20,28,6d,67,a0,70,42,a8
1019 Ja DATA 00,40,05,0f,00,4a,7a,67,01
1020 eu DATA ec,12,26,56,55,ff,58,f1,37
1021 OK DATA fe,70,76,20,4f,91,ef,07,03
1022 1a DATA e8,e8,f7,7c,72,19,e0,a4,2a
1023 7J DATA 10,2e,78,b1,1b,d8,c0,20,6d
1024 AI DATA da,37,b1,32,d9,67,12,ae,fd
1025 8N DATA de,2a,55,b0,fc,c8,df,fd,e6
1026 1v DATA 5b,2b,fd,e2,f1,36,75,93,c9
1027 VJ DATA ee,a4,da,36,68,23,40,ed,46
1028 Yr DATA 72,9e,d0,03,78,87,9b,30,70
1029 mH DATA 01,72,00,f2,01,44,47,64,4c
1030 wT DATA 01,ab,20,33,4b,6e,f7,08,08
1031 Xp DATA 35,cc,97,24,22,7a,02,fa,7d
1032 1b DATA 28,2e,2d,08,6a,33,3e,43,fa
1033 Wv DATA f2,c7,40,20,fc,9c,c5,63,86
1034 8b DATA 0a,e7,fe,7a,b1,e0,68,4b,66
1035 PZ DATA 16,30,83,e8,d2,98,07,39,50
1036 eE DATA ff,e2,29,f3,24,08,ff,e4,ea
1037 h5 DATA a7,9c,75,26,c4,91,b0,04,00
1038 1u DATA 24,0d,7e,2a,e7,0d,75,97,88
1039 AV DATA 2c,f9,fa,74,00,9c,55,8a,f9
1040 1w DATA f2,c7,40,20,fc,9c,c5,63,86
1041 ZD DATA ec,58,27,e6,0d,d0,13,1b,c0
1042 3C DATA e3,59,dc,21,7c,98,8e,67,f0
1043 Wc DATA a2,29,cd,15,1a,1f,4f,56,01
1044 8M DATA 4e,ee,f7,82,fb,88,57,ad,20
1045 uK DATA 6e,cc,20,24,48,22,ed,0a,20
1046 Jj DATA 50,0c,91,31,3e,9c,66,f0,22
1047 Fk DATA 4a,67,4f,e6,04,1e,ff,41,ee
1048 Mu DATA 01,42,90,77,20,29,57,0a,60
1049 xK DATA bc,7e,3d,64,02,7a,ff,6d,64
1050 ER DATA 48,06,67,0c,7b,4a,dc,ae,17
1051 JN DATA 05,ca,67,34,4a,87,67,20,d1
1052 1T DATA ef,86,a1,4e,02,ba,9c,a3,11
1053 h2 DATA 0a,44,a7,04,06,6a,e7,2d,d6
1054 0L DATA b9,c9,6d,dc,20,35,01,8e,42
1055 Wn DATA 10,e4,0e,00,36,7c,78,f6,60

```

Listing. »VirusControl« macht Schluß mit Viren aller Art (Fortsetzung)



1056 36 DATA 06,61,05,96,cb,ea,21,00,4a
1057 Q3 DATA 10,67,06,52,83,52,88,60,f6
1058 Jb DATA dc,7f,66,d4,62,12,22,3a,01
1059 FA DATA 60,67,04,fb,9b,42,27,04,70
1060 ma DATA 34,70,56,b6,fc,9c,61,b0,dd
1061 u6 DATA a4,8e,43,41,b0,62,10,24,08
1062 QX DATA 76,01,e2,79,ff,d0,98,04,ce
1063 bt DATA 52,b8,ff,78,f2,b0,09,bf,d7
1064 Nd DATA 0f,74,d8,0a,0a,71,0e,88,4c
1065 MO DATA 0f,c1,18,39,fe,12,20,14,77,01
1066 OD DATA 80,e5,62,8c,2e,92,35,92,73
1067 YK DATA a1,0c,48,22,3c,fc,0e,48,e7
1068 Dk DATA b4,0d,4e,ae,fe,86,4c,df,7f
1069 re DATA fe,60,da,fe,f4,b7,11,e0,0c,b5
1070 3x DATA 01,00,04,66,56,e0,8b,64,3a
1071 te DATA 20,a7,2d,2a,1e,16,e8,30,4d
1072 oW DATA 30,66,38,60,1f,23,37,64,e0
1073 h8 DATA 60,2a,5c,60,34,67,08,0c,68
1074 KD DATA 00,61,00,06,66,1a,30,28,00
1075 jb DATA 08,02,03,00,fe,07,0c,40,e5
1076 ev DATA 7b,66,0c,2f,0a,45,fa,00,a3
1077 dy DATA 14,bc,00,ff,24,5f,20,08,4e
1078 DE DATA 75,0d,f0,bf,52,e8,4c,0a,3d
1079 ky DATA 2a,00,7f,6f,13,23,37,64,22
1080 Ep DATA e0,97,81,02,ff,ff,60,60,0c
1081 vR DATA 0f,48,c1,5b,ef,17,78,6e,76
1082 CB DATA 27,90,01,bc,ae,74,75,8c,26
1083 aG DATA e8,b8,2e,6c,69,dd,2a,03,79
1084 eu DATA 00,74,69,f2,06,72,59,dc,d1
1085 TK DATA 70,18,2e,64,65,76,27,d7,9a
1086 69 DATA 1b,55,f1,05,ef,6b,0e,e0,21
1087 EB DATA 19,e0,e0,00,4b,65,80,17,e8
1088 wX DATA 64,97,4f,ec,38,2c,a5,1e,80
1089 Lj DATA 03,f2,67,03,02,31,37,3c,9b
1090 np DATA 42,4f,4f,54,42,4c,4f,43,4b
1091 9Q DATA e4,07,90,71,ed,44,00,76,9e
1092 FH DATA 0a,54,56,75,d7,03,2e,bc,57
1093 oJ DATA 3a,31,00,36,34,01,16,32,30
1094 DI DATA 30,2f,70,41,9a,9a,cb,2d,30
1095 aN DATA 19,b9,23,e1,3a,56,43,70,72
1096 vI DATA 99,ef,6f,60,6f,58,03,ea,62
1097 Ub DATA 20,a6,62,5a,3f,e5,85,56,e5
1098 oD DATA ca,4c,0e,2b,20,52,37,76,48
1099 15 DATA 49,46,54,94,cf,a0,04,98,12
1100 Mt DATA 8a,f0,72,6f,31,d3,aa,c5,05
1101 JL DATA 81,75,2f,78,0c,00,78,0d,66
1102 KN DATA 6f,e2,49,1a,f7,b9,0b,79,e4
1103 K5 DATA 72,74,c7,3f,b6,5c,66,0b,9e
1104 Q2 DATA 77,65,0b,37,98,8f,d5,3c,3f
1105 SW DATA 84,64,61,39,5e,9c,6c,15,8e
1106 NM DATA c6,91,e5,07,bc,bd,56,ed,b5
1107 t9 DATA c4,62,55,4e,09,31,c4,d6,98
1108 XF DATA 52,58,09,b0,66,ae,0b,e0,8b
1109 fJ DATA 73,6f,65,da,11,36,fb,8d,a5
1110 RQ DATA 04,a1,48,e0,8c,eb,00,1d,dc
1111 QR DATA 34,5a,54,39,e7,82,29,00,37
1112 tN DATA f7,94,dc,9c,26,62,2d,37,c5
1113 gO DATA 3b,7b,1c,67,0a,ef,3f,3f,32,e3
1114 XS DATA 38,83,69,9a,38,b5,62,26,76
1115 2D DATA 97,33,50,52,4f,47,52,41,4d
1116 10 DATA 4d,30,7e,50,49,0f,56,a2,35
1117 2B DATA 50,50,47,d2,53,b1,1a,e8,43
1118 DS DATA 4f,50,59,52,49,47,48,54,f9
1119 qA DATA 0b,4d,41,52,4b,34,26,54,45
1120 dy DATA 83,9f,77,4e,49,4b,4d,c0,80
1121 U9 DATA c2,57,88,e2,26,f6,73,30,85
1122 F1 DATA 2a,37,98,e2,5e,ab,2f,6b,f0
1123 22 DATA 44,fa,9c,51,07,92,da,2a,bc
1124 UQ DATA 20,4c,39,5f,67,77,cc,64,2d
1125 y7 DATA 55,6e,74,c3,ef,d1,69,65,c4
1126 1C DATA 95,b2,c4,67,02,27,de,6d,0d
1127 NV DATA 1f,2d,57,f3,64,6f,77,4a,4f
1128 Ia DATA c0,4d,dd,2d,45,72,72,6f,72
1129 Jb DATA dc,ef,2b,e0,47,ee,99,12,c7
1130 8a DATA 59,b8,be,47,61,64,6d,20,29
1131 Kk DATA 49,6e,68,e8,7d,74,2d,b1,90
1132 Gw DATA 72,f6,33,69,66,21,3f,e5,b8
1133 aW DATA 10,9e,2b,28,46,e6,45,44,45
1134 Ph DATA 39,9f,c2,07,28,bc,db,73,7b

1135 aB DATA 89,07,69,78,31,6c,a1,ce,b9
1136 Gd DATA 8e,d1,8b,71,3a,b2,5c,18,f0
1137 OG DATA b4,67,9c,88,94,52,20,8a,25
1138 MQ DATA 6b,5d,13,5c,e0,28,b0,0d,37
1139 vZ DATA 10,10,3c,72,7b,47,3b,4d,0a
1140 se DATA 28,78,b7,ef,16,25,80,87,ca
1141 D8 DATA 80,b7,0d,b7,51,b9,80,27,0d
1142 ZM DATA 6b,78,56,97,4d,e0,1b,6c,73
1143 De DATA 14,53,c8,1c,bc,3d,78,29,b7
1144 as DATA c3,6f,62,83,4b,e3,a2,6f,0a
1145 Ba DATA 4e,55,03,9e,08,dc,1e,bd,29
1146 Ew DATA bb,6d,20,55,43,77,1f,52,28
1147 SU DATA 20,e7,45,4e,20,6d,69,fc,a2
1148 VS DATA 0b,de,94,02,29,de,6d,ba,d3
1149 5c DATA c6,bd,dc,58,80,47,c3,84,b7
1150 5U DATA ad,b5,cf,43,4f,4e,54,b6,ab
1151 14 DATA 4c,ce,e1,18,80,c7,22,03,f0
1152 vv DATA 1e,63,c1,e0,04,2a,72,3c,db
1153 gp DATA d8,6c,ba,08,67,1a,75,6e,8a
1154 oc DATA 5f,60,00,1e,1f,8c,c1,23,40
1155 A3 DATA 0b,2a,be,f3,97,e5,c7,2c,28
1156 7y DATA de,19,c1,b8,17,73,b9,04,33
1157 FE DATA 2e,59,e5,02,25,de,51,ba,c1
1158 rO DATA c3,04,3c,33,1a,37,2d,e0,69
1159 L4 DATA of,38,46,59,80,a7,35,e0,3a
1160 58 DATA 2d,00,0c,ec,18,08,f0,a0,61
1161 Tm DATA 02,2b,64,9e,27,6b,b4,9e,a3
1162 vK DATA 61,68,61,64,2d,66,0e,2c,20
1163 rd DATA 44,86,fb,6e,b5,66,fe,c6,69
1164 z3 DATA de,d1,12,e0,09,2b,9c,99,8b
1165 vt DATA e3,0c,5e,8b,02,2b,de,6d,ba
1166 Iv DATA 3d,44,38,bc,38,80,47,28,eb
1167 fd DATA 4c,a9,37,4b,53,27,ce,3f,8d
1168 27 DATA 90,95,4e,b3,08,14,bc,10,05
1169 IU DATA ba,46,bb,7d,38,e0,25,00,e1
1170 gN DATA 00,01,03,3b,40,16,0a,00,5a
1171 QX DATA a5,dd,ce,7b,52,52,4f,52,9d
1172 gH DATA 1b,6e,66,1f,e9,30,16,8f,ac
1173 zH DATA 62,8b,35,cb,b0,05,ef,14,2e
1174 H4 DATA e0,2c,ec,39,04,41,7b,54,55
1175 oB DATA 4e,47,00,20,53,43,48,52,3a
1176 1x DATA 42,5a,55,47,52,49,46,62,20
1177 XK DATA 61,75,66,26,fe,f4,9a,04,dc
1178 hR DATA 85,f6,80,4c,65,73,db,84,7f
1179 EE DATA 0e,ce,a2,97,0e,ae,4e,01,2d
1180 Ye DATA 22,cf,a8,33,01,20,2f,9a,44
1181 Jf DATA 46,30,6c,c3,77,69,72,6b,ea
1182 nR DATA 9b,45,aa,fc,d7,72,03,1f,63
1183 aD DATA 72,69,69,bb,7d,46,ae,0e,ce
1184 zM DATA 49,01,62,78,39,60,4a,5c,c4
1185 KY DATA 51,22,f0,7e,5e,4e,45,49,4e
1186 Sh DATA db,eb,66,45,28,e0,9a,65,84
1187 Fd DATA 31,2e,33,f1,6e,8d,73,4a,02
1188 Gz DATA 66,25,01,63,47,8c,03,26,de
1189 h7 DATA ae,b9,69,09,2d,da,78,1e,d0
1190 gT DATA 7e,00,f0,52,6f,75,4d,6e,22
1191 40 DATA fd,7e,c0,01,bc,00,2d,da,6e
1192 fK DATA 17,72,2d,53,69,6d,75,6a,61
1193 zy DATA fb,96,7b,93,60,c0,2b,f0,80
1194 E1 DATA 6c,f5,88,73,3a,20,ed,7b,61
1195 Rp DATA d9,65,20,17,65,24,80,68,75
1196 aH DATA 65,6c,ad,35,e7,69,a4,78,91
1197 MX DATA cf,74,cc,84,e4,42,d3,a5,b6
1198 og DATA b3,2e,6a,33,f9,4e,c0,8b,90
1199 NQ DATA 55,26,eb,8c,02,0f,c2,ce,dd
1200 tS DATA 68,84,2e,88,4c,ba,01,19,78
1201 nV DATA 1a,04,78,0a,10,78,53,c2,e0
1202 M2 DATA 80,67,02,0f,00,1a,ef,2b,39
1203 JK DATA e7,23,37,ca,5e,57,8a,31,65
1204 Hd DATA 6b,88,b5,78,20,15,e3,e4,15
1205 pD DATA 1a,56,f7,2a,11,3a,d0,20,3f
1206 eg DATA 8c,c3,c4,00,2f,ad,c6,49,c8
1207 Ou DATA d2,ae,17,0d,c3,66,12,56,49
1208 f1 DATA 52,ea,2d,45,4a,54,46,19,7e
1209 Tn DATA 4e,c0,68,45,52,53,55,43,48
1210 w5 DATA f5,4d,46,c5,74,75,72,7a,67
1211 Ot DATA 65,66,61,68,9b,1f,01,29,62
1212 yH DATA 65,73,6c,73,ef,22,4d,49,57
1213 yS DATA 41,f0,17,20,21,21,20,ec,48
1214 as DATA 03,5e,00,ed,69,70,69,f6,85
1215 AF DATA f5,72,22,57,65,c8,02,f2,c8
1216 N8 DATA ef,03,2a,22,0d,d0,49,19,2f
1217 N9 DATA 78,3b,e7,86,b2,74,73,cb,28
1218 a6 DATA fc,fa,48,8a,96,22,b7,1d,e0

1219 QW DATA cd,89,2d,19,53,75,6e,9e,d2
1220 cJ DATA 55,b6,e7,1a,62,69,c3,04,30
1221 3B DATA bc,71,b5,0a,88,59,4b,e8,d9
1222 5Q DATA 38,8c,55,53,f3,f5,80,68,77
1223 Cv DATA 40,c4,cb,6b,20,13,e0,2d,5a
1224 Ev DATA c2,75,74,7a,fe,f5,80,17,0c
1225 or DATA 20,4b,37,d3,e0,53,79,1b,ab
1226 Zw DATA 8a,32,6d,1b,15,9d,f3,02,f0
1227 FH DATA b8,7e,77,2d,9a,e1,ca,ea,8a
1228 X8 DATA 30,e0,3c,57,59,02,84,3a,e9
1229 Cr DATA 57,23,07,6e,69,6e,67,2d,c5
1230 RL DATA 97,4b,e0,31,1d,96,42,5b,3e
1231 IR DATA bf,2d,56,d9,5b,75,73,43,7d
1232 FT DATA 67,72,6f,6c,db,58,03,5e,5b
1233 UM DATA ad,72,0a,b9,02,bc,2a,a1,2a
1234 gn DATA f0,b4,e7,66,2a,2a,e7,f1,73
1235 c3 DATA 79,6d,dd,51,94,e1,cb,00,06
1236 39 DATA bc,68,3a,15,4e,6f,9c,73,74
1237 LJ DATA 28,46,d1,e5,84,e0,a8,7d,c5
1238 7d DATA 74,61,6a,64,db,b7,64,24,89
1239 kt DATA e7,03,0f,50,aa,9a,c0,1a,08
1240 Vq DATA 81,3a,ec,b2,03,31,f8,3c,03
1241 rC DATA d0,62,05,a0,ab,c6,6c,65,99
1242 FV DATA 7f,e1,bc,00,bc,1a,b8,42,6f
1243 D1 DATA 6f,74,42,79,b7,6b,20,2d,3e
1244 Dm DATA 5c,1f,c2,69,72,6d,b9,23,e0
1245 Vo DATA 07,a2,c9,38,a0,cc,25,48,54
1246 9A DATA ea,20,46,61,72,62,3c,94,1e
1247 Qy DATA e0,41,4a,33,ae,9c,93,97,41
1248 xM DATA 78,81,2f,d7,70,66,41,17,de
1249 3o DATA 02,5c,b9,09,34,6d,81,2f,b4
1250 y8 DATA e7,01,16,bb,05,88,72,23,60
1251 98 DATA 4d,5b,e0,2f,dc,f9,ba,63,60
1252 fD DATA 00,b4,02,25,ef,e5,05,4f,5f
1253 n6 DATA c0,63,e0,02,9c,b5,15,b1,0f
1254 9n DATA 84,80,e1,d1,0c,13,bc,05,0c
1255 FD DATA 72,2b,68,52,bd,73,ae,07,9a
1256 2p DATA f3,ef,b0,12,f0,12,16,c8,38
1257 Bz DATA 00,23,1b,82,ea,00,58,3d,43
1258 2U DATA 4b,05,77,cc,e0,11,5b,20,64
1259 z5 DATA 37,81,06,e1,05,30,de,63,38
1260 S6 DATA 17,54,68,9f,33,90,73,04,ac
1261 OF DATA 41,78,81,30,2c,e7,01,3b,ab
1262 XC DATA 1f,a7,02,72,54,0c,c2,b0,30
1263 kt DATA bc,e2,3e,00,0e,2f,15,33,e8
1264 J2 DATA e7,04,d2,c3,c0,7c,3b,60,e7
1265 y8 DATA 2a,e3,54,14,4c,a0,3d,0b,30
1266 4S DATA a4,3c,0f,d8,1d,c2,34,40,00
1267 D8 DATA d0,b9,eb,61,e0,30,85,99,01
1268 Ne DATA 0b,bc,05,72,a4,22,70,b3,65
1269 4s DATA c1,88,01,da,bd,2f,21,26,2b
1270 48 DATA 06,03,f0,f0,60,86,80,b1,00
1271 4J DATA a5,11,eb,c0,30,f4,f3,8d,1d
1272 3e DATA 88,18,c0,c4,4d,20,53,0c,30
1273 4e DATA 1e,03,07,74,ae,34,f0,38,97
1274 dz DATA d3,a9,09,31,de,b7,58,03,72
1275 w1 DATA 41,b1,0e,34,c4,51,95,f1,ba
1276 80 DATA c0,53,82,59,10,57,10,eb,40
1277 rX DATA 48,9b,87,40,af,0b,94,bc,03
1278 LX DATA 76,a3,74,a0,74,4d,a0,27,0b
1279 p3 DATA 31,bc,3c,0f,d8,45,f9,35,a0
1280 g4 DATA 39,27,69,30,ac,02,31,e4,af
1281 rP DATA 87,3c,00,af,40,0e,51,08,00
1282 1K DATA 5b,7d,56,e0,59,d9,23,0f,c0
1283 ak DATA 23,35,f8,01,98,00,79,00,ae
1284 10 DATA 8a,9f,12,66,09,c4,04,37,90
1285 yb DATA eb,42,32,34,9c,bb,29,7c,bc
1286 92 DATA 27,76,45,40,4a,19,6a,c3,36
1287 ZG DATA cc,f3,89,dd,ee,4c,ae,50,02
1288 +2 DATA b5,e1,1c,3d,b1,60,db,8a,42
1289 Fm DATA 29,48,f0,14,36,24,f6,7e,b8
1290 33 DATA 02,28,00,0e,97,27,db,9b,09
1291 Q1 DATA b0,96,4a,c0,37,6c,33,c4,28
1292 3A DATA 43,71,0d,78,4c,ac,39,ab,75
1293 QP DATA fb,81,c7,64,d9,04,do,ae,a0
1294 b2 DATA b2,ed,2c,9b,03,d7,5b,35,80
1295 1x DATA 30,b7,c0,e9,d9,1c,0a,44,07
1296 21 DATA 9b,0d,ca,64,07,05,b4,5b,e9
1297 Bl DATA 07,37,e0,bc,dd,fc,2b,2d,0c
1298 fe DATA ad,70,a4,ae,06,00,03,ee,3c
1299 fe DATA 55,6a,b8,67,54,33,3f,ed,38
1300 W6 DATA ae,01,2f,ab,0c,02,14,a7,04
1301 1W DATA b2,a8,04,48,09,8a,18,c0,41
1302 4o DATA ef,02,c0,82,0d,fa,a3,37,00

1303 Jb DATA 01,83,52,37,10,c8,0c,04,01
1304 vb DATA 02,38,ed,a0,ff,00,0d,7d,ae
1305 Cp DATA 4e,2b,06,26,20,82,fa,02,ff
1306 8H DATA 4c,99,05,04,90,05,20,53,3b
1307 e5 DATA 10,c8,8a,12,a0,3c,60,3f,00
1308 KL DATA 04,d0,26,32,ff,4d,02,40,01
1309 Xz DATA 17,ff,f9,60,3e,9d,45,08,00
1310 2H DATA 01,74,6f,70,61,7a,2e,66,6f
1311 eS DATA 6a,74,ed,6f,38,a8,00,09,02
1312 Zz DATA 64,7e,3d,df,01,a5,7a,40,52
1313 wV DATA 9e,dc,6c,e7,03,98,49,78,eb
1314 Re DATA 76,da,1a,83,82,81,25,e9,92
1315 4K DATA 82,75,aa,15,77,74,30,53,bb
1316 p4 DATA 80,56,5e,1a,53,77,66,11,48
1317 tv DATA da,94,70,b6,4e,63,68,ae,3c
1318 gW DATA e3,96,61,26,1a,33,23,43,6f
1319 HZ DATA 0e,30,2d,60,9c,8b,27,72,75
1320 wV DATA 71,05,0b,48,76,14,eb,4d,61
1321 lry DATA 6b,63,ae,16,2e,b7,42,30,8e
1322 8h DATA ea,2b,90,19,77,ac,65,1b,61
1323 Ah DATA 44,4b,ab,ed,f2,62,75,08,39
1324 qE DATA b4,ae,69,96,70,37,61,62,28
1325 oI DATA ae,9f,e5,0f,37,62,69,20,9e
1326 7a DATA 85,33,1e,6d,0c,7d,60,1d,0e
1327 aK DATA 0c,52,d2,c0,75,ba,96,b3,88
1328 CQ DATA 55,b9,7a,53,74,43,0a,b8,78
1329 Rg DATA 4a,56,65,ca,ed,83,33,6d,34
1330 JH DATA dc,e3,91,84,2c,5a,72,89,38
1331 E4 DATA 43,61,75,db,29,c2,2e,37,6d
1332 eJ DATA a7,71,2e,74,4e,e5,a8,ed,b8
1333 iW DATA 1d,1b,56,60,17,73,3e,bf,b4
1334 Dx DATA 00,32,e6,14,76,dc,a0,ab,18
1335 lE DATA 27,0b,87,14,0e,ca,68,31,07
1336 LX DATA 74,f9,02,eb,00,ff,ec,e8,fa
1337 dr DATA 93,4b,6a,bb,1b,d1,ba,b9,a4
1338 Id DATA 6f,76,5c,12,e7,92,48,65,61
1339 lS DATA 64,ba,01,00,69,de,28,45,6e
1340 l8 DATA 71,78,03,75,18,10,e3,00,4e
1341 BB DATA b0,99,b4,d2,e4,86,bb,36,2a
1342 J4 DATA ea,36,b5,23,f3,80,7a,54,61
1343 oK DATA 73,6b,71,fe,72,69,1e,1b,76
1344 l4 DATA 57,45,78,4b,86,15,00,b9,81
1345 T9 DATA 61,eb,76,d5,dc,82,46,65,c4
1346 Df DATA 1c,95,53,69,67,5b,55,66,16
1347 3F DATA 6c,36,63,49,62,46,2e,d1,65
1348 HX DATA 94,f8,43,70,b6,c8,0d,77,31
1349 Kz DATA 94,40,e7,31,ed,32,dc,d4,73
1350 Ro DATA 31,e7,80,70,6c,79,4d,73,f9
1351 cE DATA 14,b9,41,62,44,46,69,9d,9d
1352 gR DATA 04,32,d6,95,40,70,57,42,07
1353 4m DATA 4f,6a,64,f6,b6,97,2e,2b,02
1354 zF DATA f7,ca,49,74,76,57,6e,63,88
1355 CY DATA 07,97,99,81,75,6d,b8,9a,0e
1356 Ae DATA 09,1b,ee,92,a8,a8,3b,d2,8c
1357 Kb DATA 39,6c,6f,11,5e,44,63,76,69
1358 v7 DATA 9b,97,44,d0,10,39,e6,0c,53
1359 V4 DATA 5c,85,05,d5,6d,88,cd,6b,24
1360 Xh DATA e6,57,61,fc,e3,98,53,62,90
1361 w7 DATA 0a,16,8b,1c,41,64,64,dd,c3
1362 4n DATA 5d,6d,80,45,dd,91,52,65,73
1363 lJ DATA 6f,76,08,59,90,a4,49,6f,dc
1364 EJ DATA 69,25,bf,2c,4d,61,79,69,6b
1365 lx DATA 68,2e,99,93,50,75,f2,68,84
1366 86 DATA 06,82,52,61,77,44,6f,46,6d
1367 gA DATA e0,bf,c0,47,ed,c2,43,98
1368 5N DATA 04,0e,54,79,39,4f,66,4d,65
1369 BR DATA 64,84,4f,50,72,6f,63,65,64
1370 Xf DATA 75,aa,47,8c,96,61,63,d1,26
1371 Yp DATA 88,03,0f,0e,4f,52,6e,cd,56
1372 LP DATA 72,7a,79,2f,97,2f,bc,6b,00
1373 pe DATA 3c,29,01,da,f0,e8,64,10,1e
1374 wG DATA 83,ca,c0,c3,2a,f0,76,e1,4f
1375 rt DATA 55,54,bd,77,61,5d,a3,2c,c0
1376 XC DATA 2a,f0,80,61,2d,44,4d,41,20
1377 Qd DATA 62,65,f6,87,64,65,e5,3c,f1
1378 Up DATA 4e,8b,c0,cb,96,7f,18,80,97
1379 op DATA 10,43,f0,b4,08,bc,6c,a1,84
1380 OE DATA 41,49,39,78,83,e7,43,6f,70
1381 QK DATA 70,e2,36,f4,0b,07,f0,40,38
1382 eT DATA f0,54,76,84,c0,6b,7a,c8,02
1383 67 DATA 06,09,61,73,74,6c,fc,63,6b
1384 7y DATA 65,c0,8f,55,e0,05,42,6c,69
1385 Hq DATA 74,ca,b7,20,0a,4e,ca,67,32
1386 qE DATA 0c,06,e0,1d,37,30,78,83,21

1387 b4 DATA 31,bc,e1,10,32,5e,aa,03,34
1388 eZ DATA 20,41,75,64,cf,8f,64,48,0f
1389 CV DATA 89,20,4b,15,03,61,c9,33,18
1390 TG DATA 63,e1,b0,0a,73,bc,0d,2e,fe
1391 rO DATA 49,4e,2d,70,75,66,66,68,ff
1392 cC DATA 8a,76,6f,6c,6c,5e,22,81,35
1393 kE DATA 20,44,ef,6b,53,79,6e,63,68
1394 61 DATA 72,04,bf,69,73,61,74,c0,0d
1395 Ag DATA 73,77,c8,c0,74,62,15,43,49
1396 jD DATA 41,2d,42,2e,f8,de,70,61,6e
1397 eg DATA 73,69,6f,6a,50,6f,72,a9,7e
1398 S1 DATA 16,0d,ab,c0,69,cb,6e,65,37
1399 6W DATA 20,53,6f,66,74,77,61,72,65
1400 75 DATA f2,73,52,51,70,00,45,78,65
1401 Fg DATA 63,4c,69,62,2e,f1,1f,2d,36
1402 fW DATA 22,4e,4d,49,2d,49,6e,74,65
1403 8U DATA 70,72,75,70,74,f7,00,fe,03
1404 ct DATA 1b,20,20,20,0d,06,da,f0,80
1405 3a DATA 16,72,bc,06,3a,66,69,6c,65
1406 oe DATA 6e,61,6d,65,9f,35,38,f0,04
1407 mb DATA 03,b0,d6,40,88,18,00,b1,0b
1408 T2 DATA bc,e0,58,02,48,00,c5,3c,b1
1409 WE DATA ed,c1,09,08,7a,65,2d,04,4f
1410 Sh DATA 69,c0,23,5d,68,32,0b,67,04
1411 w1 DATA 30,70,80,a7,ed,51,71,bc,b5
1412 41 DATA 04,0f,9a,36,e0,ed,e1,03,bc
1413 VC DATA 9a,38,67,81,80,97,85,97,f0
1414 fJ DATA 30,60,05,1e,16,6e,c2,83,a4
1415 Ba DATA c0,5b,bf,25,85,2b,19,7f,fe
1416 Fw DATA 97,bf,6f,f0,3f,f9,0c,c6,fd
1417 4J DATA e2,2c,75,32,68,45,66,89,b4
1418 qT DATA 80,5e,04,fc,49,32,31,ce,18
1419 yW DATA a7,01,b7,93,2b,41,03,18,2f
1420 f3 DATA 02,07,fe,5a,0a,02,bc,6e,00
1421 Og DATA 99,38,0c,14,83,f8,78,5b,b0
1422 3D DATA c0,d3,95,05,81,84,c0,0d,04
1423 dH DATA fb,eb,7a,4b,00,06,6f,82,6e
1424 gZ DATA 27,70,93,80,c0,23,c0,1a,34
1425 hf DATA f0,80,f6,88,90,c0,ef,eb,0d
1426 YW DATA 03,f0,22,e2,9e,07,5c,08,c0
1427 oE DATA 02,b6,16,b1,b6,70,c0,5c,65
1428 QO DATA e7,28,18,f0,06,8f,26,04,05
1429 mX DATA c0,5e,0e,3a,44,5c,f9,57,c3
1430 gT DATA 16,03,05,1e,1b,dc,0f,9f,4c
1431 lF DATA 7c,6b,ca,00,0f,10,03,60,c1
1432 tZ DATA 53,87,59,15,ac,b3,00,c0,43
1433 XL DATA 82,99,0f,11,3e,36,61,05,a8
1434 FE DATA e0,31,d7,ec,5b,ee,c1,01,af
1435 co DATA 45,b0,1e,3c,24,fe,bb,09,ab
1436 6F DATA 41,02,6f,08,9e,c0,61,d0,59
1437 eE DATA 50,b2,f0,bc,60,a6,01,e7,37
1438 fE DATA f1,40,0a,02,60,bc,01,e4,63
1439 db DATA 94,56,80,0a,5a,72,9c,3b,b0
1440 lu DATA c0,e1,4b,15,e1,3d,98,30,c0
1441 H1 DATA b3,e7,45,f8,41,2b,c8,0a,e2
1442 G8 DATA 1b,b2,07,07,3c,a6,b6,1b,bc
1443 wp DATA 27,61,a0,c0,8b,31,60,82,e7
1444 2R DATA 20,04,b2,c4,e0,dd,91,0a,ca
1445 Yf DATA 90,6b,84,9b,e1,88,df,0b,0f
1446 lD DATA 5a,68,7f,ff,bf,b7,0b,0d,1f
1447 yd DATA 68,fe,ff,df,b4,03,3f,74,82
1448 Kp DATA 19,81,8a,25,c7,93,48,6c,48
1449 N1 DATA f0,18,84,81,53,e8,b7,2a,6c
1450 cM DATA 03,ff,81,fc,9d,40,c0,fe,3f
1451 p1 DATA 0d,80,03,7f,80,b3,c0,8f,09
1452 3s DATA 1a,78,08,7e,01,fc,f4,7e,03
1453 Yy DATA fa,01,68,62,5a,93,b4,ff,01
1454 lW DATA f8,7e,07,c0,fc,3e,01,fe,fc
1455 nn DATA b7,6e,00,6a,96,c0,8f,80,3f,ed
1456 Az DATA 67,7a,80,e7,c4,e2,01,4d,78
1457 eg DATA 40,0b,1e,d0,80,97,16,97,f0
1458 sk DATA ca,e2,1d,30,6c,c0,03,80,2d
1459 6Q DATA 79,cf,4b,1e,d0,88,07,34,e1
1460 ue DATA 05,e9,e0,7d,0f,78,10,0c,78
1461 vT DATA 50,03,1e,00,03,3c,5e,2f,41
1462 VA DATA ea,03,cf,dc,4c,2c,0a,4d,41,ea
1463 v3 DATA 01,ee,0c,34,e0,39,b0,c2,55
1464 LS DATA 0d,e0,c0,c0,cf,7b,ab,1e,e0
1465 2G DATA fc,c0,39,40,1d,6e,86,1a,50
1466 Rb DATA 80,c0,37,9b,ee,e0,f0,8d,0f
1467 Cs DATA 6a,87,1e,99,e0,d3,ec,c2,00
1468 CN DATA af,4a,2d,48,35,94,3c,83,11
1469 7H DATA c0,1f,87,2d,8f,20,c3,6a,80
1470 RK DATA 0d,fc,1f,21,2f,78,e0,91,13

1471 kJ DATA e0,09,08,d9,dc,c4,8d,00,37
1472 xA DATA 80,ac,57,b2,8f,82,60,82,b3
1473 nO DATA 3e,34,61,c2,96,75,15,0a,78
1474 dK DATA 13,36,ec,01,7e,68,4a,e4,00
1475 y4 DATA 3c,5f,c0,e7,48,ae,0f,c0,7f
1476 zn DATA f0,dd,21,94,23,21,0f,34,1b
1477 ex DATA c2,a3,03,f0,58,16,9f,c0,04
1478 8t DATA c1,9e,65,7d,00,68,19,38,f0
1479 kq DATA 83,62,10,e0,ad,c5,fa,d0,1e
1480 Xs DATA e4,dc,85,01,7e,3f,9e,7b,85
1481 EK DATA 45,96,00,0a,1a,83,6d,b8,fd
1482 up DATA 65,8e,f0,ff,99,05,ef,df,c4
1483 vt DATA c5,72,18,33,10,7f,6a,af,04
1484 xH DATA 89,00,c0,9e,86,44,03,99,91
1485 n8 DATA 58,a2,fe,f4,43,b3,1d,06,09
1486 2i DATA 7f,7e,07,61,47,48,e0,91,c3
1487 21 DATA 80,60,7e,b0,c0,73,8f,e7,e1
1488 AP DATA e5,08,09,7f,bc,17,06,78,06
1489 9B DATA 39,52,60,26,08,7e,78,43,c7
1490 DB DATA 68,04,ae,82,8f,c0,58,7e,12
1491 V7 DATA 0f,b8,ca,f1,e2,87,20,0c
1492 X8 DATA 3a,e4,c6,b0,0d,ff,99,58,63
1493 oK DATA 60,c0,83,9b,47,0d,12,d6,a0
1494 4S DATA 03,3f,fe,9e,0f,62,c0,85,c1,11
1495 5a DATA 3c,65,98,61,7f,60,13,24,f0
1496 LC DATA 4c,e8,01,f0,b0,0b,0c,c1,bb
1497 xA DATA 0d,87,ea,3f,f8,60,80,87,57
1498 7L DATA 02,00,41,5f,b7,82,c7,57,87
1499 46 DATA ff,84,2c,03,93,82,c7,ae,95
1500 T4 DATA 83,77,01,ff,6b,06,03,75,f0
1501 gX DATA e1,e0,7f,7f,19,a7,ff,8f,1f,fe
1502 bE DATA b4,63,fc,60,bd,14,3f,bc,19
1503 P2 DATA 8f,e7,f1,6c,26,0f,e3,f1,fc
1504 DA DATA bb,7f,8e,19,8f,e3,f8,ec,04
1505 tM DATA f2,90,41,5f,e2,a5,c0,3d,f3
1506 aP DATA f9,70,83,f7,f1,fe,7e,3f,07
1507 Fd DATA be,4d,74,cf,e3,fc,fc,c0,02
1508 Vn DATA f8,ae,0e,1d,5a,5b,0a,c9,de
1509 lB DATA 0c,0f,ff,f2,80,54,c8,a5,e0
1510 te DATA a0,f0,5b,42,00,04,6f,8c,07
1511 eE DATA e3,9c,fc,41,33,68,40,fc,3f
1512 0u DATA bf,8d,81,a5,f8,88,81,f8,cd
1513 tY DATA 66,f0,e1,32,07,ff,9b,5d,f8
1514 8J DATA 24,05,f0,dd,6c,06,ef,84,9f
1515 lG DATA 36,19,01,14,da,7f,f0,62,0e
1516 8o DATA 16,75,c7,04,e0,30,83,0f,f0
1517 10 DATA ac,f8,23,02,05,03,5e,2d,19
1518 lO DATA 0f,7f,9f,c2,7f,80,3f,87,b1
1519 0e DATA ff,00,1f,e0,3f,0f,80,f8,1f
1520 OV DATA 87,c0,3f,c0,ff,f8,00,1a,c0
1521 np DATA ff,01,f9,f0,07,01,af,93,59
1522 wP DATA 52,7f,b2,80,4a,bc,0e,0f,3f
1523 Ga DATA e8,c0,f0,1e,f0,e0,62,01,1e
1524 f8 DATA 93,09,c0,ff,80,b9,28,4e,c0
1525 2g DATA 1d,c5,d0,68,d2,03,1a,fc,80
1526 lN DATA 46,3c,e0,05,0f,68,c0,03,1a
1527 gh DATA f0,c3,12,d5,10,54,e0,ff,ff
1528 oH DATA ff,f0,4e,d4,21,08,c0,2a,47
1529 oe DATA 11,00,23,51,5f,e6,ae,e1,e0
1530 cI DATA 06,3b,55,19,2e,53,82,6a,00
1531 bF DATA a2,25,00,4e,e4,a0,b2,00,68
1532 BU DATA 09,00,ad,c0,9e,24,07,d5,c0
1533 nr DATA ba,46,0e,2a,8e,00,e6,1c,ea
1534 LB DATA d4,4a,fe,1e,04,d4,a0,08,0e
1535 5S DATA 1c,33,22,a8,41,26,0d,aa,81
1536 3J DATA 01,40,1c,44,d4,a0,60,06,d5
1537 oE DATA 01,78,44,0e,7c,6a,b4,50,83
1538 4t DATA 2a,01,a1,45,06,30,e4,a0,36
1539 Rq DATA 06,3a,35,46,a8,42,54,0d,5a
1540 Ig DATA 6a,5a,50,83,66,1a,6a,d4,a0
1541 JN DATA 7e,06,84,35,88,a8,41,55,06
1542 Jh DATA 9c,90,83,a6,1a,ae,d4,a0,c0
1543 2n DATA 06,95,06,e0,40,16,06,f8,e0
1544 8C DATA a0,02,06,0c,35,72,a8,41,8c
1545 3g DATA 0d,96,aa,07,f6,00,07,92,35
1546 V7 DATA ba,a8,02,08,d4,2c,0a,4a,36
1547 DR DATA 40,19,a5,m0,eb,a8,02,00,01
1548 Sq DATA 00,00,03,f2,f6,90,80,00,00
1549 qK DATA 00,03,eb,00,00,1a,3d,00,00
1550 Na DATA 03,f2

(C) 1989 MET
**Listing. »VirusControl« macht
Schluß mit Viren aller Art (Schluß)**

Geräteinspektion

Wollten Sie auch schon in einem Programm testen, ob ein externes Laufwerk oder eine Festplatte angeschlossen sind? Ein Kinderspiel mit »TestDev.h«.

Immer, wenn ein Programm zum Speichern auf ein anderes als das interne Laufwerk zugreifen möchte, sollte getestet werden, ob das entsprechende Laufwerk tatsächlich vorhanden ist. Bei den maximal vier Diskettenlaufwerken stellt dies ja noch kein Problem dar.

```
#include <devices/trackdisk.h>
struct IOExtTD testReq;
LONG DRIVE1 = GADGDISABLED;
if(!OpenDevice("trackdisk.device", 1L, &testReq, 0L))
{
    CloseDevice(&testReq);
    DRIVE1 = 0L;
}
```

In diesem Beispiel wird für eine spätere Gadget-Abfrage das Flag für die Darstellung als »Ghost«-Gadget gesetzt und, wenn das Laufwerk vorhanden ist, wieder gelöscht (0L in der letzten Zeile). Für die vier Laufwerke müssen in der IF-Abfrage Werte zwischen 0L und 3L eingesetzt werden.

Alles schön und gut, aber eine Festplatte oder das RAM kann so nicht getestet werden, da sie nicht über das trackdisk.device angesprochen werden. Da aber jedes angeschlossene und gemountete

Device beim Start des Amiga in eine Liste eingetragen wird, ist es am einfachsten, in dieser Liste nach dem entsprechenden Device zu suchen.

Die Amiga-Entwickler haben es aber nicht leicht gemacht, die relevanten Daten zu finden. Ein Programm muß sich zuerst durch mehrere Systemlisten hangeln. Der Weg führt über DOSBase, RootNode und DosInfo hin zur DeviceList. Deren Definitionen kann der interessierte Programmierer der Include-Datei »libraries/dosexters.h« entnehmen.

Dort angekommen, kann die in der Include-Datei »libraries/filehandler.h« definierte Struktur ausgelesen werden. Hierbei offenbart sich ein weiteres Hindernis. Einige der Zeiger sind BCPL-Zeiger (BPTR), die erst umgerechnet werden müssen. Dazu dient die Definition in Zeile 15 in Listing 1, die das BCPL-Format durch Multiplikation mit dem Wert 4 in das übliche C-Adressformat umrechnet. In der while-Schleife wird das Element »dn_Type« durchsucht und mit der Konstanten »DLT_DEVICE« verglichen. Nur wenn der Vergleich positiv ist, wird der Eintrag weiterbearbeitet. Da hier noch andere Knotentypen stehen, wir aber nur nach echten Devices suchen, werden die anderen Typen so übergangen.

In den Zeilen 25 bis 29 wird der so gefundene Name in einen C-String umgewandelt. Hierzu eine kurze Erläuterung: Bei einem BCPL-String gibt das erste Byte die Länge des Strings an, die folgenden Bytes sind die einzelnen Zeichen. Ein C-String dagegen wird durch ein Nullbyte abgeschlossen. Der so gewonnene String wird in Zeile 33 mit dem beim Aufruf der Unterroutine übergebenen Ausdruck verglichen. Stimmt der Vergleich, kehrt die Unterroutine mit dem Ergebnis TRUE zurück. Beim Aufruf von TestDev muß das zu suchende Device ohne Doppelpunkt übergeben werden. Wie das Ergebnis, das TestDev liefert, später vom Hauptprogramm ausgewertet wird, hängt ganz allein von Ihnen ab. Auf alle Fälle verfügen Sie nun über eine Kontrollmöglichkeit, die das Vorhandensein eines Devices anzeigt.

Dieter Mayer/kn

Programname:	TestDev.h
Bemerkung:	Nur zum Einbinden
Programmautor:	Dieter Mayer

1	hK0 /* Include-Datei "TestDev.h" Tester, ob
2	Y5 vorgegebenes Device angeschlossen ist. *
3	bN #ifndef LIBRARYES_LC32
4	-f3 #include <libraries/filehandler.h>
5	MIO #endif
6	dh #ifndef LIBRARYES_DOSBASE
7	7y3 #include <libraries/dosexters.h>
8	PLO #endif
9	g TestDev(device)
10	7D char *device;
11	7a
12	3Q3 char DeviceName[10], *zeige,
13	cm BOOL erg = FALSE
14	in" zähl,
15	7g #define GetBtr(x) ((BPTR) (x) << 2)
16	bV struct DeviceInfo *devinfo, *next;
17	k extern struct DosLibrary *DosBase;
18	9R struct DosInfo *info;
19	3v struct RootNode *root;
20	70 root = DosBase->RootNode;
21	7I info = GetBtr(root->Info);
22	7Y devinfo = GetBtr(info->devinfo);
23	7 while devinfo;
24	kn
25	AC6 next = GetBtr(devinfo->dn_Next);
26	6e if devinfo->dn_Type == DLT_DEVICE
27	7q
28	HH9 zeige = GetBtr(devinfo->dn_Name);
29	7m zähl = *zeige;
30	7b if zähl > 10 zähl = 10;
31	7J strcpy(DeviceName, zeige, zähl);
32	F9 DeviceName[zähl] = 0;
33	7g if strcmp(DeviceName, device) == 0 erg = TRUE;
34	736
35	eh devinfo = next;
36	73
37	7X return erg;
38	720,
39	(C) 1989 M&T

Listing 2.
»TestDev.h« testet das Vorhandensein von Devices (Geräten).

Programname:	Demo.c
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	CC Demo +L -S LN Demo.o -LC32
Programmautor:	Dieter Mayer

1	700 *
2	7D /* Demoprogramm zur Testen, ob ein bestimmtes
3	7C Device angeschlossen ist.
4	7C *
5	7F 12.88.88, Dieter Mayer
6	71 *
7	7k #include "TestDev.h"
8	7C if main(argc, argv)
9	71
10	73 BYTE * "00", BYTE * "01", BYTE * "02",
11	73 BYTE * "03", BYTE * "04", BYTE * "05",
12	73 BYTE * "06", BYTE * "07", BYTE * "08",
13	73 BYTE * "09", BYTE * "0A", BYTE * "0B",
14	73 BYTE * "0C", BYTE * "0D", BYTE * "0E",
15	73 BYTE * "0F", BYTE * "10", BYTE * "11",
16	73 BYTE * "12", BYTE * "13", BYTE * "14",
17	73 BYTE * "15", BYTE * "16", BYTE * "17",
18	73 BYTE * "18", BYTE * "19", BYTE * "1A",
19	73 BYTE * "1B", BYTE * "1C", BYTE * "1D",
20	73 BYTE * "1E", BYTE * "1F", BYTE * "20",
21	73 BYTE * "21", BYTE * "22", BYTE * "23",
22	73 BYTE * "24", BYTE * "25", BYTE * "26",
23	73 BYTE * "27", BYTE * "28", BYTE * "29",
24	73 BYTE * "2A", BYTE * "2B", BYTE * "2C",
25	73 BYTE * "2D", BYTE * "2E", BYTE * "2F",
26	73 BYTE * "30", BYTE * "31", BYTE * "32",
27	73 BYTE * "33", BYTE * "34", BYTE * "35",
28	73 BYTE * "36", BYTE * "37", BYTE * "38",
29	73 BYTE * "39", BYTE * "3A", BYTE * "3B",
30	73 BYTE * "3C", BYTE * "3D", BYTE * "3E",
31	73 BYTE * "3F", BYTE * "40", BYTE * "41",
32	73 BYTE * "42", BYTE * "43", BYTE * "44",
33	73 BYTE * "45", BYTE * "46", BYTE * "47",
34	73 BYTE * "48", BYTE * "49", BYTE * "4A",
35	73 BYTE * "4B", BYTE * "4C", BYTE * "4D",
36	73 BYTE * "4E", BYTE * "4F", BYTE * "50",
37	73 BYTE * "51", BYTE * "52", BYTE * "53",
38	73 BYTE * "54", BYTE * "55", BYTE * "56",
39	73 BYTE * "57", BYTE * "58", BYTE * "59",
40	73 BYTE * "5A", BYTE * "5B", BYTE * "5C",
41	73 BYTE * "5D", BYTE * "5E", BYTE * "5F",
42	73 BYTE * "60", BYTE * "61", BYTE * "62",
43	73 BYTE * "63", BYTE * "64", BYTE * "65",
44	73 BYTE * "66", BYTE * "67", BYTE * "68",
45	73 BYTE * "69", BYTE * "6A", BYTE * "6B",
46	73 BYTE * "6C", BYTE * "6D", BYTE * "6E",
47	73 BYTE * "6F", BYTE * "70", BYTE * "71",
48	73 BYTE * "72", BYTE * "73", BYTE * "74",
49	73 BYTE * "75", BYTE * "76", BYTE * "77",
50	73 BYTE * "78", BYTE * "79", BYTE * "7A",
51	73 BYTE * "7B", BYTE * "7C", BYTE * "7D",
52	73 BYTE * "7E", BYTE * "7F", BYTE * "80",
53	73 BYTE * "81", BYTE * "82", BYTE * "83",
54	73 BYTE * "84", BYTE * "85", BYTE * "86",
55	73 BYTE * "87", BYTE * "88", BYTE * "89",
56	73 BYTE * "8A", BYTE * "8B", BYTE * "8C",
57	73 BYTE * "8D", BYTE * "8E", BYTE * "8F",
58	73 BYTE * "90", BYTE * "91", BYTE * "92",
59	73 BYTE * "93", BYTE * "94", BYTE * "95",
60	73 BYTE * "96", BYTE * "97", BYTE * "98",
61	73 BYTE * "99", BYTE * "9A", BYTE * "9B",
62	73 BYTE * "9C", BYTE * "9D", BYTE * "9E",
63	73 BYTE * "9F", BYTE * "A0", BYTE * "A1",
64	73 BYTE * "A2", BYTE * "A3", BYTE * "A4",
65	73 BYTE * "A5", BYTE * "A6", BYTE * "A7",
66	73 BYTE * "A8", BYTE * "A9", BYTE * "AA",
67	73 BYTE * "AB", BYTE * "AC", BYTE * "AD",
68	73 BYTE * "AE", BYTE * "AF", BYTE * "B0",
69	73 BYTE * "B1", BYTE * "B2", BYTE * "B3",
70	73 BYTE * "B4", BYTE * "B5", BYTE * "B6",
71	73 BYTE * "B7", BYTE * "B8", BYTE * "B9",
72	73 BYTE * "BA", BYTE * "BB", BYTE * "BC",
73	73 BYTE * "BD", BYTE * "BE", BYTE * "BF",
74	73 BYTE * "C0", BYTE * "C1", BYTE * "C2",
75	73 BYTE * "C3", BYTE * "C4", BYTE * "C5",
76	73 BYTE * "C6", BYTE * "C7", BYTE * "C8",
77	73 BYTE * "C9", BYTE * "CA", BYTE * "CB",
78	73 BYTE * "CC", BYTE * "CD", BYTE * "CE",
79	73 BYTE * "CF", BYTE * "D0", BYTE * "D1",
80	73 BYTE * "D2", BYTE * "D3", BYTE * "D4",
81	73 BYTE * "D5", BYTE * "D6", BYTE * "D7",
82	73 BYTE * "D8", BYTE * "D9", BYTE * "DA",
83	73 BYTE * "DB", BYTE * "DC", BYTE * "DD",
84	73 BYTE * "DE", BYTE * "DF", BYTE * "E0",
85	73 BYTE * "E1", BYTE * "E2", BYTE * "E3",
86	73 BYTE * "E4", BYTE * "E5", BYTE * "E6",
87	73 BYTE * "E7", BYTE * "E8", BYTE * "E9",
88	73 BYTE * "EA", BYTE * "EB", BYTE * "EC",
89	73 BYTE * "ED", BYTE * "EE", BYTE * "EF",
90	73 BYTE * "F0", BYTE * "F1", BYTE * "F2",
91	73 BYTE * "F3", BYTE * "F4", BYTE * "F5",
92	73 BYTE * "F6", BYTE * "F7", BYTE * "F8",
93	73 BYTE * "F9", BYTE * "FA", BYTE * "FB",
94	73 BYTE * "FC", BYTE * "FD", BYTE * "FE",
95	73 BYTE * "FF", BYTE * "00", BYTE * "01",
96	73 BYTE * "02", BYTE * "03", BYTE * "04",
97	73 BYTE * "05", BYTE * "06", BYTE * "07",
98	73 BYTE * "08", BYTE * "09", BYTE * "0A",
99	73 BYTE * "0B", BYTE * "0C", BYTE * "0D",
100	73 BYTE * "0E", BYTE * "0F", BYTE * "10",
101	73 BYTE * "11", BYTE * "12", BYTE * "13",
102	73 BYTE * "14", BYTE * "15", BYTE * "16",
103	73 BYTE * "17", BYTE * "18", BYTE * "19",
104	73 BYTE * "1A", BYTE * "1B", BYTE * "1C",
105	73 BYTE * "1D", BYTE * "1E", BYTE * "1F",
106	73 BYTE * "20", BYTE * "21", BYTE * "22",
107	73 BYTE * "23", BYTE * "24", BYTE * "25",
108	73 BYTE * "26", BYTE * "27", BYTE * "28",
109	73 BYTE * "29", BYTE * "2A", BYTE * "2B",
110	73 BYTE * "2C", BYTE * "2D", BYTE * "2E",
111	73 BYTE * "2F", BYTE * "30", BYTE * "31",
112	73 BYTE * "32", BYTE * "33", BYTE * "34",
113	73 BYTE * "35", BYTE * "36", BYTE * "37",
114	73 BYTE * "38", BYTE * "39", BYTE * "3A",
115	73 BYTE * "3B", BYTE * "3C", BYTE * "3D",
116	73 BYTE * "3E", BYTE * "3F", BYTE * "40",
117	73 BYTE * "41", BYTE * "42", BYTE * "43",
118	73 BYTE * "44", BYTE * "45", BYTE * "46",
119	73 BYTE * "47", BYTE * "48", BYTE * "49",
120	73 BYTE * "4A", BYTE * "4B", BYTE * "4C",
121	73 BYTE * "4D", BYTE * "4E", BYTE * "4F",
122	73 BYTE * "50", BYTE * "51", BYTE * "52",
123	73 BYTE * "53", BYTE * "54", BYTE * "55",
124	73 BYTE * "56", BYTE * "57", BYTE * "58",
125	73 BYTE * "59", BYTE * "5A", BYTE * "5B",
126	73 BYTE * "5C", BYTE * "5D", BYTE * "5E",
127	73 BYTE * "5F", BYTE * "60", BYTE * "61",
128	73 BYTE * "62", BYTE * "63", BYTE * "64",
129	73 BYTE * "65", BYTE * "66", BYTE * "67",
130	73 BYTE * "68", BYTE * "69", BYTE * "6A",
131	73 BYTE * "6B", BYTE * "6C", BYTE * "6D",
132	73 BYTE * "6E", BYTE * "6F", BYTE * "70",
133	73 BYTE * "71", BYTE * "72", BYTE * "73",
134	73 BYTE * "74", BYTE * "75", BYTE * "76",
135	73 BYTE * "77", BYTE * "78", BYTE * "79",
136	73 BYTE * "7A", BYTE * "7B", BYTE * "7C",
137	73 BYTE * "7D", BYTE * "7E", BYTE * "7F",
138	73 BYTE * "80", BYTE * "81", BYTE * "82",
139	73 BYTE * "83", BYTE * "84", BYTE * "85",
140	73 BYTE * "86", BYTE * "87", BYTE * "88",
141	73 BYTE * "89", BYTE * "8A", BYTE * "8B",
142	73 BYTE * "8C", BYTE * "8D", BYTE * "8E",
143	73 BYTE * "8F", BYTE * "90", BYTE * "91",
144	73 BYTE * "92", BYTE * "93", BYTE * "94",
145	73 BYTE * "95", BYTE * "96", BYTE * "97",
146	73 BYTE * "98", BYTE * "99", BYTE * "9A",
147	73 BYTE * "9B", BYTE * "9C", BYTE * "9D",
148	73 BYTE * "9E", BYTE * "9F", BYTE * "A0",
149	73 BYTE * "A1", BYTE * "A2", BYTE * "A3",
150	73 BYTE * "A4", BYTE * "A5", BYTE * "A6",
151	73 BYTE * "A7", BYTE * "A8", BYTE * "A9",
152	73 BYTE * "AA", BYTE * "AB", BYTE * "AC",
153	73 BYTE * "AD", BYTE * "AE", BYTE * "AF",
154	73 BYTE * "B0", BYTE * "B1", BYTE * "B2",
155	73 BYTE * "B3", BYTE * "B4", BYTE * "B5",
156	73 BYTE * "B6", BYTE * "B7", BYTE * "B8",
157	73 BYTE * "B9", BYTE * "BA", BYTE * "BB",
158	73 BYTE * "BC", BYTE * "BD", BYTE * "BE",
159	73 BYTE * "BF", BYTE * "C0", BYTE * "C1",
160	73 BYTE * "C2", BYTE * "C3", BYTE * "C4",
161	73 BYTE * "C5", BYTE * "C6", BYTE * "C7",
162	73 BYTE * "C8", BYTE * "C9", BYTE * "CA",
163	73 BYTE * "CB", BYTE * "CC", BYTE * "CD",
164	73 BYTE * "CE", BYTE * "CF", BYTE * "D0",
165	73 BYTE * "D1", BYTE * "D2", BYTE * "D3",
166	73 BYTE * "D4", BYTE * "D5", BYTE * "D6",
167	73 BYTE * "D7", BYTE * "D8", BYTE * "D9",
168	73 BYTE * "DA", BYTE * "DB", BYTE * "DC",
169	73 BYTE * "DD", BYTE * "DE", BYTE * "DF",
170	73 BYTE * "E0", BYTE * "E1", BYTE * "E2",
171	73 BYTE * "E3", BYTE * "E4", BYTE * "E5",
172	73 BYTE * "E6", BYTE * "E7", BYTE * "E8",
173	73 BYTE * "E9", BYTE * "EA", BYTE * "EB",
174	73 BYTE * "EC", BYTE * "ED", BYTE * "EE",
175	73 BYTE * "EF", BYTE * "F0", BYTE * "F1",
176	73 BYTE * "F2", BYTE * "F3", BYTE * "F4",
177	73 BYTE * "F5", BYTE * "F6", BYTE * "F7",
178	73 BYTE * "F8", BYTE * "F9", BYTE * "FA",
179	73 BYTE * "FB", BYTE * "FC", BYTE * "FD",
180	73 BYTE * "FE", BYTE * "FF", BYTE * "00",
181	73 BYTE * "01", BYTE * "02", BYTE * "03",
182	73 BYTE * "04", BYTE * "05", BYTE * "06",
183	73 BYTE * "07", BYTE * "08", BYTE * "09",
184	73 BYTE * "0A", BYTE * "0B", BYTE * "0C",
185	73 BYTE * "0D", BYTE * "0E", BYTE * "0F",
186	73 BYTE * "10", BYTE * "11", BYTE * "12",
187	73 BYTE * "13", BYTE * "14", BYTE * "15",
188	73 BYTE * "16", BYTE * "17", BYTE * "18",
189	73 BYTE * "19", BYTE * "1A", BYTE * "1B",
190	73 BYTE * "1C", BYTE * "1D", BYTE * "1E",
191	73 BYTE * "1F", BYTE * "20", BYTE * "21",
192	73 BYTE * "22", BYTE * "23", BYTE * "24",
193	73 BYTE * "25", BYTE * "26", BYTE * "27",
194	73 BYTE * "28", BYTE * "29", BYTE * "2A",
195	73 BYTE * "2B", BYTE * "2C", BYTE * "2D",
196	73 BYTE * "2E", BYTE * "2F", BYTE * "30",
197	73 BYTE * "31", BYTE * "32", BYTE * "33",
198	73 BYTE * "34", BYTE * "35", BYTE * "36",
199	73 BYTE * "37", BYTE * "38", BYTE * "39",
200	73 BYTE * "3A", BYTE * "3B", BYTE * "3C",
201	73 BYTE * "3D", BYTE * "3E", BYTE * "3F",
202	73 BYTE * "40", BYTE * "41", BYTE * "42",
203	73 BYTE * "43", BYTE * "44", BYTE * "45",
204	73 BYTE * "46", BYTE * "47", BYTE * "48",
205	73 BYTE * "49", BYTE * "4A", BYTE * "4B",
206	73 BYTE * "4C", BYTE * "4D", BYTE * "4E",
207	73 BYTE * "4F", BYTE * "50", BYTE * "51",
208	73 BYTE * "52", BYTE * "53", BYTE * "54",
209	73 BYTE * "55", BYTE * "56", BYTE * "57",
210	73 BYTE * "58", BYTE * "59", BYTE * "5A",
211	73 BYTE * "5B", BYTE * "5C", BYTE * "5D",
212	73 BYTE * "5E", BYTE * "5F", BYTE * "60",
213	73 BYTE * "61", BYTE * "62", BYTE * "63",
214	73 BYTE * "64", BYTE * "65", BYTE * "66",
215	73 BYTE * "67", BYTE * "68", BYTE * "69",
216	73 BYTE * "6A", BYTE * "6B", BYTE * "6C",
217	73 BYTE * "6D", BYTE * "6E", BYTE * "6F",
218	73 BYTE * "70", BYTE * "71", BYTE * "72",
219	73 BYTE * "73", BYTE * "74", BYTE * "75",
220	73 BYTE * "76", BYTE * "77", BYTE * "78",
221	73 BYTE * "79", BYTE * "7A", BYTE * "7B",
222	73 BYTE * "7C", BYTE * "7D", BYTE * "7E",
223	73 BYTE * "7F", BYTE * "80", BYTE * "81",
224	73 BYTE * "82", BYTE * "83", BYTE * "84",
225	73 BYTE * "85", BYTE * "86", BYTE * "87",
226	73 BYTE * "88", BYTE * "89", BYTE * "8A",
227	73 BYTE * "8B", BYTE * "8C", BYTE * "8D",
228	73 BYTE * "8E", BYTE * "8F", BYTE * "90",
229	73 BYTE * "91", BYTE * "92", BYTE * "93",
230	73 BYTE * "94", BYTE * "95", BYTE * "96",
231	73 BYTE * "97", BYTE * "98", BYTE * "99",
232	73 BYTE * "9A", BYTE * "9B", BYTE * "9C",
233	73 BYTE * "9D", BYTE * "9E", BYTE * "9F",
234	73 BYTE * "A0", BYTE * "A1", BYTE * "A2",
235	73 BYTE * "A3", BYTE * "A4", BYTE * "A5",
236	73 BYTE * "A6", BYTE * "A7", BYTE * "A8",
237	73 BYTE * "A9", BYTE * "AA", BYTE * "AB",
238	73 BYTE * "AC", BYTE * "AD", BYTE * "AE",
239	73 BYTE * "AF", BYTE * "B0", BYTE * "B1",
240	73 BYTE * "B2", BYTE * "B3", BYTE * "B4",
241	73 BYTE * "B5", BYTE * "B6", BYTE * "B7",
242	73 BYTE * "B8", BYTE * "B9", BYTE * "BA",
243	73 BYTE * "BB", BYTE * "BC", BYTE * "BD",
244	73 BYTE * "BE", BYTE * "BF", BYTE * "C0",
245	73 BYTE * "C1", BYTE * "C2", BYTE * "C3",
246	73 BYTE * "C4", BYTE * "C5", BYTE * "C6",
247	73 BYTE * "C7", BYTE * "C8", BYTE * "C9",
248	73 BYTE * "CA", BYTE * "CB", BYTE * "CC",
249	73 BYTE * "CD", BYTE * "CE", BYTE * "CF",
250	73 BYTE * "D0", BYTE * "D1", BYTE * "D2",
251	73 BYTE * "D3", BYTE * "D4", BYTE * "D5",
252	73 BYTE * "D6", BYTE * "D7", BYTE * "D8",
253	73 BYTE * "D9", BYTE * "DA", BYTE * "DB",
254	73 BYTE * "DC", BYTE * "DD", BYTE * "DE",
255	73 BYTE * "DF", BYTE * "E0", BYTE * "E1",
256	73 BYTE * "E2", BYTE * "E3", BYTE * "E4",
257	73 BYTE * "E5", BYTE * "E6", BYTE * "E7",
258	73 BYTE * "E8", BYTE * "E9", BYTE * "EA",
259	73 BYTE * "EB", BYTE * "EC", BYTE * "ED",
260	73

Ein großes Plus des Amiga sind die Fenster. Viele Informationen lassen sich so gleichzeitig darstellen. Doch wie kommt man schnell und einfach an das hinterste Fenster? »Shuffle« gibt Ihnen die Antwort.

EIn Programm nimmt Ihnen die Arbeit ab, Fenster nach hinten zu klicken oder zu schließen. Außerdem steckt in dem Listing »Shuffle« hochinteressanter Stoff: Es handelt sich nämlich um einen Input-Handler. »Shuffle« (mischen) hängt sich hinter das Input-Device und vor die anderen Programme. Drückt man die Taste <Help>, legt es das nächste Fenster nach oben, über dem der Mauszeiger steht. Sie müssen also nicht alle Fenster »durchblättern«, bis Sie zum gewünschten gelangen. Außer <Help> erkennt Shuffle noch die Tastenkombination <rechte Amiga Esc>. Damit schalten Sie das Programm aus. Es wird komplett aus dem System herausgenommen.

Doch wie macht man so etwas? Anhand des Listings erläutern wir es. Einige Dinge sind wichtig, wenn man ein Programm wie Shuffle schreiben möchte. Zunächst stellt sich die Frage, wie man einen Hintergrundprozeß startet. Bei Verwendung des Aztec-C-Compilers verwendet man hierfür zusätzlich zu den benötigten C-Bibliotheken noch »ldetach.o«. Damit dies funktioniert, benötigt das Programm noch zusätzliche Informationen. Die Variablen finden Sie in den Zeilen 76 bis 79.

Das Programm läuft also im Hintergrund. Als nächste Aufgabe steht das Auswerten von Tastendrücken an. Zunächst brauchen wir eine Struktur, die die Adresse unseres Tasks sowie die Signale und die entsprechenden Nummern enthält. Diese Struktur heißt »GLOBAL_DATA« und wird in den Zeilen 15 bis 22 definiert. Die verwendete Struktur hat dann den Namen »global«. Das Eintragen der Werte in die Struktur erfolgt im Hauptprogramm in den Zeilen 109, 119, 123, 124 und 128. Um die Task-Adresse zu ermitteln, verwenden wir die Funktion FindTask(0L) (Zeile 109). Für die Belegung der Signale gibt es die Funktion AllocSignal() (Zeilen 119 und 124). Der Parameter -1 besagt, daß wir keine spezielle Tasknummer wünschen. Das Bit wird dann einfach durch Verschieben des Wertes 1 gesetzt (Zeilen 123 und 128). Die Struktur »global« ist jetzt vollständig gefüllt.

Die nächste erforderliche Struktur ist »handlerStuff« (Zeile 105). Es handelt sich dabei um eine Interrupt-Struktur. Diese wird nun in den Zeilen 137 bis 139 gefüllt. Als erstes kommt der Zeiger auf die schon definierte Struktur »global«. Es folgt ein Zeiger auf die Interface-Routine (»HandlerInterface«). Sie ist in Assembler geschrieben, da die Parameter in den Registern A0 und A1 geliefert werden. Damit kann aber die eigentliche Interrupt-Routine nichts anfangen, da sie in C geschrieben ist. Die Registerinhalte werden auf den Stack gelegt und die Funktion »myhandler« aufgerufen. Die letzte der Zeilen setzt die Priorität des Interrupts. In den Zeilen 144 bis 146 binden wir unseren Interrupt-Handler ins System ein.

In der »Endlos«-Schleife erfolgt die Auswertung eines aufgetretenen Signals. Mit der Funktion Wait() legen wir Shuffle schlafen, bis ein Signal empfangen wird. Erhalten wir das Signal zum Beenden, verlassen wir die Schleife und geben die vom System angeforderten Bibliotheken, Signale etc. wieder frei. Danach endet unser Programm. Empfangen wir jedoch das Signal für Aktion, holen wir das nächste Fenster nach vorn.

Die eben erwähnten Signale erhalten wir von der Routine »myhandler« (Zeile 29 bis 71). Hier erfolgt die Auswertung der gedrückten Taste. Handelt es sich um die Tastenkombination <rechte Amiga Esc>, schicken wir das Signal zum Beenden. Ist es die <Help>-Taste, geben wir das Signal zur Aktion. Trifft keiner der beiden Fälle zu, geben wir den Zeiger auf die Event-Liste zurück. Dadurch kann der nächste Handler arbeiten, als wäre nichts passiert.

Natürlich können Sie statt des Blätterns der Fenster auch andere Dinge erledigen. Achten Sie jedoch unbedingt darauf, daß Ihre Routine so wenig Zeit wie irgend möglich benötigt. Das ist wichtig, damit das System nicht unnötig belastet wird. Falls Ihnen die Tastenbelegung von Shuffle nicht gefällt, ändern Sie diese in den Zeilen 12 bis 13.

Shuffle arbeitet mit den Fenstern des aktuellen Bildschirms. Dadurch ist es universell einsetzbar, auch in Programmen, die einen eigenen Screen öffnen.

René Beaupoll/kn

BITTE UMBL

Programmname:	Shuffle
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.6
Aufrufer:	CC Shuffle +C +D +L LN Shuffle.o ldetach.o32 -LCL32

Programmierer: Steffen Auerbach

```

1 510 /*****
2 *F /* Shuffle Version 2.0 */
3 *x /* Autor: Steffen Auerbach */
4 *F /* Philipp-Kels-Str. 1 */
5 *T /* 64. Haus 1 */
6 *C /*****
7 * * * * * INCLUDE FILES * * * * *
8 #include <functions.h>
9 #include <devices/input.h>
10 #include <interrupt/interruptbase.h>
11 #define ESCAPEKEY 0x5f /* Tastencode für Esc-Taste
12 #define HELPKEY 0x5f /* Tastencode für Help-Taste
13 #define GLOBAL_VARIABLES * * * * *
14 typedef struct
15 {
16     struct Task *TaskToSig, /* Adresse des warte
17     rden Tasks */
18     ULONG QuitSig, /* Signal um Handler
19     zu beenden */
20     ULONG QuitSigNr, /* Signal Nr. für Fr
21     eeSignal() */
22     ULONG ActionSig, /* Signal zum Blätte
23     ren */
24     ULONG ActionSigNr, /* Signal Nr. für Fr
25     eeSignal() */
26     GLOBAL_DATA *global;
27     struct MsgPort *inputPort;
28     struct Task *inputRequestBlock;
29     /* Die Handleroutine, die vom HandlerInterface aufgerufen
30     wird */
31     /* InputEvent *myhandler(ev, gdp)
32     struct inputEvent *ev, /* Pointer auf die Eventliste
33     */
34     register GLOBAL_DATA *gdp, /* Alle benötigten Informatio
35     nen */
36     register struct inputEvent *ev, *laste,
37     /* Die Eventliste durchsuchen um zu sehen,
38     */
39     /* ob die richtige Taste gedrückt wurde
40     */
41     for (ep = ev, laste = NULL; ep != NULL; ep = ep->ie_Next
42     event)
43     if (ep->ie_Class == (ECLASS_PAWKEY) &&
44         ep->ie_Code == ESCAPEKEY &&
45         (ep->ie_Qualifier & QUALIFIER_COMMAND))
46     {
47         /* Falls wir den Event bearbeiten können => */
48         /* Event aus der Liste löschen */
49         if (laste == NULL)
50             ev = ep->ie_NextEvent;
51         else

```



```

117 ke if ((InputRequestBlock < CreateStdIO(inputDevPort)) == 0
118
119 116 #f
120 117 #26
121 118 #3
122 119 #g
123 120 #s
124 121 #j6
125 122 #3
126 123 #2
127 124 #6
128 125 #x2
129 126 #x0
130 127 #j3
131 128 #B
132 129 #J
133
134 130 #V
135 131 #*6
136 132 #d3
137 133 #7
138 134 #Z
139 135 #t6
140 136 #j3
141 137 #X
142 138 #f
143 139 #B
144 140 #p
145 141 #s
146 142 #26
147 143 #3
148 144 #
149 145 #6
150 146 #*
151 147 #a
152 148 #v
153 149 #Q6
154 150 #u
155 151 #i
156 152 #W9
157 153 #*6
158 154 #t
159 155 #p9
160 156 #q
161 157 #f
162 158 #j
163 159 #-
164 160 #P
165 161 #V
166 162 #d
167 163 #o
168 164 #p
169 165 #2
170 166 #s
171 167 #7
172 168 #f
173 169 #n0
174 170 #M9
175 171 #A
176 172 #g0
177 173 #67
178 174 #9
179 175 #C1
180 176 #4
181 177 #x
182 178 #J6
183 179 #03
184 180 #*
185 181 #B
186 182 #06
187 183 #D
188 184 #S
189 185 #U0
190
191 (C) 1989 M&I

```

AMIGA-SOURCE LEFT 3

Das Arbeiten mit numerischen Feldern in Amiga-Basic verschlingt viel Zeit. Das muß nicht sein. Ein kleiner Umweg macht Ihre Programme wesentlich schneller.

von René Beaupoil und Matthias Rustler

Es gibt zwei Wege mit dem Auto vom Münchner Norden in den Süden zu kommen.

— Der kurze Weg quer durch die überfüllte Stadt.
— Der längere Weg über den Autobahnring, auf dem man (außer in der Urlaubszeit) flott vorwärtskommt.

Im zweiten Fall kommen Sie schneller ans Ziel. Die Basic-Subprogramme von »Felder« machen Ihre Programme etwas länger, beschleunigen sie aber erheblich. Der Trick an der Sache ist, daß sich Zeichenketten (Strings) schneller manipulieren lassen als numerische Felder. Doch wie wandelt man diese Felder in Strings um? Das Betriebssystem hilft uns.

Der benutzte Befehl stammt aus der »exec.library« und heißt »CopyMem()«. Er kopiert einen beliebig langen Speicherbereich in einen anderen. Die Subprogramme kopieren also numerische Felder in eine Zeichenkette und nehmen dort die gewünschte Manipulation vor. Anschließend kopieren Sie den String wieder zurück.

Um die Routinen zu verwenden, müssen Sie sie zu Ihrem Basic-Programm hinzufügen. Dies geschieht mit dem Befehl MERGE.

Der schnelle

Außerdem muß im selben Verzeichnis, in dem Amiga-Basic und das Programm steht, auch die Datei »exec.bmap« vorhanden sein. Diese Datei finden Sie auf Ihrer Extras-Diskette im Verzeichnis »BasicDemos«.

Es stehen Ihnen sechs Routinen zur Verfügung, die nun eingehend besprochen werden.

□ FastLoad

FastLoad field%(), »Name«, puffer

Diese Routine lädt Werte aus einer sequentiellen Datei in das angegebene Feld. Der Parameter »field%()« ist der Name des numerischen Feldes, das Sie benutzen. Mit »Name« gibt man den kompletten Zugriffspfad für die Datei an. »puffer« bestimmt, wie groß der Speicherbereich ist, der zum Laden verwendet wird. Je größer dieser Wert ist, um so schneller wird geladen. Andererseits verbraucht dies auch mehr Speicher.

□ FastSave

FastSave field%(), »Name«, puffer

Dieses Kommando ist das Gegenstück zum Befehl »FastLoad«. Die Parameter haben dieselbe Bedeutung.

□ Init

Init field%(), anfang, ende, wert

Hiermit initialisieren Sie das Feld »field%()«. Normalerweise sind alle Feldelemente 0. Sie können jedoch einem beliebigen Teil einen bestimmten Wert (»wert«) zuweisen. Dies geschieht vom Element

Programmname: Felder

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: Amiga-Basic 1.2

Bemerkung: siehe Text

Programmautor: Matthias Rustler

```
1 100 LIBRARY "exec.library"
2 20 CLEAR,30000
3 30 DIM p%(1500)
4 40 PRINT "Initialisieren mit der neuen Routine"
5 50 v=TIMER
6 60 Init p%(),0,1500,300
7 70 vn=TIMER-v
8 80 PRINT "Initialisieren normal"
9 90 v=TIMER
10 100 FOR i=0 TO 1500 p%(i)=300 NEXT
11 110 vo=TIMER-v
12 120 GOSUB PrintFeLd
13 130 PRINT "Mit Feld-Routinen zirka 'vo/vn' mal schneller"
14 140 PRINT
15 150 PRINT "Einfügen mit der neuen Routine"
16 160 v=TIMER
17 170 Insert p%(),750,55
18 180 vn=TIMER-v
19 190 PRINT "Einfügen normal"
20 200 v=TIMER
21 210 FOR i=1500 TO 751 STEP-1:p%(i)=p%(i-1):NEXT
22 220 p%(750)=55
23 230 vo=TIMER-v
24 240 GOSUB PrintFeLd
25 250 PRINT "Mit Feld-Routinen zirka 'vo/vn' mal schneller"
26 260 PRINT
27 270 PRINT "Löschen mit der neuen Routine"
28 280 v=TIMER
29 290 Del p%(),10,12
30 300 vn=TIMER-v
31 310 PRINT "Löschen normal"
32 320 v=TIMER
33 330 FOR i=10 TO 1498:p%(i)=p%(i+2):NEXT
34 340 vo=TIMER-v
35 350 GOSUB PrintFeLd
36 360 PRINT "Mit Feld-Routinen zirka 'vo/vn' mal schneller"
37 370 PRINT
38 380 PRINT "Suchen mit der neuen Routine"
39 390 v=TIMER
40 400 Search p%(),0,55,x%
41 410 vn=TIMER-v
42 420 PRINT "Suchen normal"
```

```
43 430 v=TIMER
44 440 FOR i=0 TO 1500 :p%(i)=55 THEN rrr
45 450 NEXT i
46 460 rrr=
47 470 vo=TIMER-v
48 480 PRINT "55 hat den Index ",x%
49 490 PRINT "Mit Feld-Routinen zirka 'vo/vn' mal schneller"
50 500 PRINT
51 510 PRINT "Drücken Sie den linken Mausknopf"
52 520 WHILE MOUSE(0)=0:WEND
53 530 CLS
54 540 ERASE p%
55 550 g=6022
56 560 DIM p%(g)
57 570 FOR x=0 TO 300 STEP 10
58 580 LINE (x,0)-(300-x,150),RND*3
59 590 NEXT
60 600 GOTO (0,0)-(300,150),p%
61 610 LOCATE 20,5
62 620 PRINT "Speichern mit der neuen Routine"
63 630 v=TIMER
64 640 FastSave p%(),"Bild",4000
65 650 vn=TIMER-v
66 660 LOCATE 21,5
67 670 PRINT "Speichern normal"
68 680 v=TIMER
69 690 OPEN "Bild2" FOR OUTPUT AS 1
70 700 FOR i=0 TO 6022
71 710 PRINT #1,VALSTR(p%(i),
72 720 NEXT i
73 730 CLOSE 1
74 740 vo=TIMER-v
75 750 LOCATE 22,5
76 760 PRINT "Zirka 'vo/vn' mal schneller"
77 770 LOCATE 23,40
78 780 PRINT "Laden mit der neuen Routine"
79 790 v=TIMER
80 800 FastLoad p%(),"Bild",4000
81 810 vn=TIMER-v
82 820 LOCATE 21,40
83 830 PRINT "Laden normal"
84 840 v=TIMER
85 850 OPEN "Bild2" FOR INPUT AS 1
86 860 FOR i=0 TO 6022
87 870 a$=INPUT$ 2,#1
88 880 p%(i)=VAL(a$)
89 890 NEXT i
90 900 CLOSE 1
91 910 vo=TIMER-v
92 920 LOCATE 22,40
93 930 PRINT "Zirka 'vo/vn' mal schneller"
94 940 GOTO (320,0),p%
```


Der Umweg

»anfang« bis zum Element »ende«. Wollen Sie keine Werte ändern, lautet der Aufruf: Init feld%,0,0,0

☐ Insert Insert feld%(),position,wert

Das Einfügen von neuen Werten in das Feld geschieht schnell und einfach. Mit »position« geben Sie den Feldindex an, an dem das neue Element stehen soll. »wert« stellt den einzufügenden Zahlenwert dar. Die nachfolgenden Feldelemente werden nach hinten verschoben.

☐ Del Del feld%(),anfang,ende

Löscht beliebig viele Feldelemente. Die dahinterliegenden Elemente werden um »ende-anfang« Stellen nach vorne verschoben. »anfang« legt das erste zu löschende Feld fest. Mit »ende« bestimmt man das letzte zu löschende Element.

☐ Search Search feld%(),anfang,wert,ergebnis%

Oft sucht man nach einem Wert und benötigt den entsprechenden Feldindex. Damit die Routine flexibel ist und nicht nur den ersten Wert findet, legen Sie mit »anfang« den Index fest, ab dem gesucht wird. Der zu suchende Wert wird im Parameter »wert« übergeben. »ergebnis%« liefert zwei Werte als Ergebnis:

— Bei erfolgreicher Suche enthält »ergebnis%« den Index des Feldelements, das den gesuchten Wert enthält.

— Konnte der Wert nicht gefunden werden, liefert »ergebnis%« -1. In der vorliegenden Version arbeiten die Subprogramme mit Integervariablen (2 Byte). Wenn Sie andere Datentypen verwenden wollen, müssen Sie folgende Änderungen vornehmen:

— Das Typkennzeichen der Variablen muß geändert werden. Das jetzt vorhandene Zeichen »%« muß durch eines der folgenden ersetzt werden

&: lange Ganzzahl, 4 Byte

!: Fließkommazahl, einfach genau, 4 Byte

#: Fließkommazahl, doppelt genau, 8 Byte

— »VL%« muß die Zahl der Bytes je Feldelement enthalten.

— Die MKL\$-Funktionen müssen, dem Typ entsprechend, in MKL\$ (lange Ganzzahl), MKS\$ (einfach genaue Fließkommazahl) oder MKD\$ (doppelt genaue Fließkommazahl) geändert werden.

Da Zeichenketten in Basic maximal 32767 Zeichen groß sein dürfen, sind die Routinen auch auf diese Länge beschränkt. Die maximale Anzahl der Feldelemente ergibt sich aus 32767/VL%. Ausnahmen bilden die Routinen »FastLoad« und »FastSave«. Hier sind bis zu 32767 Feldelemente erlaubt.

Listing 1 zeigt Ihnen den Geschwindigkeitsvorteil. Dies sind natürlich keine fixen Werte, da es von verschiedenen Faktoren abhängt. Aber Faktoren bis zu 20 treten durchaus auf. Die Zeilen 2 bis 107 können Sie durch Ihr eigenes modifiziertes Programm ersetzen.

Einen Nachteil hat das schnelle Autofahren natürlich: Der Benzinverbrauch ist höher. Bei dem Programm ist es analog: Der Speicherbedarf bei Verwendung der Routinen ist höher. Aber so lange Sie genügend Sprit haben... kn

```

95 BP LIBRARY CLOSE
96 ID END
97 GA PrintFeld:
98 nE PRINT "Index ",
99 FN FOR a=0 TO 15
100 DX2 PRINT USING "#####",a,
101 InO NEXT
102 O7 PRINT :PRINT "Wert: ";
103 JR FOR a=0 TO 15
104 wE2 PRINT USING "#####";p(a ,
105 mrO NEXT
106 4h PRINT
107 5h RETURN
108 6C SUB FastLoad (f%(),File$,puffl%) STATIC
109 20 VL%=2
110 Bx IF puffl%<1 THEN puffl%=1024
111 ja OPEN File$ FOR INPUT AS 1
112 ap2 lok%=0
113 GD lges%=LOF(1)
114 r5 IF (UBOUND(f%)+1)*VL% < lges% THEN ERROR 9
115 z8 WHILE lok%<lges%
116 684 IF lges%-lok%<puffl% THEN puffl%=lges%-lok%
117 xW z8=INPUT$(puffl%,1)
118 LF CALL CopyMem(SADD(z8),VARPTR(f%(0))+lok%,puffl%)
119 CF lok%=lok%+puffl%
120 NR2 WEND
121 hA z8=""
122 mWO CLOSE 1
123 IJ END SUB
124 I1 SUB FastSave (f%(),File$,puffl%) STATIC
125 I0 VL%=2
126 RD IF puffl%<1 THEN puffl%=1024
127 Pb OPEN File$ FOR OUTPUT AS 1
128 u52 lok%=0
129 2D lges%=(UBOUND(f%)+1)*VL%
130 H1 z8=STRING$(puffl%,CHR$(0))
131 pO WHILE lok%<lges%
132 M44 IF lges%-lok%<puffl% THEN puffl%=lges%-lok%
133 Ih CALL CopyMem(VARPTR(f%(0))+lok%,SADD(z8),puffl%)
134 IY PRINT #1,LEFT$(z8,puffl%);
135 SY lok%=lok%+puffl%
136 dP2 WEND
137 KQ z8=""
138 2mO CLOSE 1
139 EJ END SUB
140 8Q SUB Init (f%(),von%,bis%,wert%) STATIC
141 YW VL%=2
142 J5 IF bis%=0 THEN bis%=(UBOUND(f%)+1)*VL%
143 m1 IF von%>bis% THEN SWAP von%,bis%
144 D6 Test=f%(von%)
145 CE Test=f%(bis%)
146 Ue l%=(bis%-von%+1)*VL%


```

```

147 cQ z8=VNIS wert%.
148 OT WHILE l%>2<VL%
149 r72 z8=z8+z8
150 rFO WEND
151 e1 z8=z8-LEFT$(z8,l%-LEN z8,
152 JU CALL CopyMem(SADD(z8),VARPTR(f%(von%)),l%)
153 ng z8=""
154 WY END SUB
155 T2 SUB Insert (f%(),ps%,wert%) STATIC
156 n1 VL%=2
157 ar Test=f%(ps%)
158 Bu IF ps%<UBOUND(f%) THEN
159 ae2 l%=(UBOUND(f%)-ps%+1)*VL%
160 Ib z8=STRING$(l%,CHR$(0))
161 mu CALL CopyMem(VARPTR(f%(ps%)+SADD(z8)),l%
162 Jh CALL CopyMem(SADD(z8),VARPTR(f%(ps%+1)),l%-VL%)
163 ZS0 END IF
164 yr z8=""
165 w2 f%(ps%)=wert%
166 Ik END SUB
167 KQ SUB Del (f%(),von%,bis%) STATIC
168 ex VL%=2
169 wW IF bis%=0 THEN bis%=(UBOUND(f%)+1)*VL%
170 D9 IF von%>bis% THEN SWAP von%,bis%
171 eX Test=f%(von%)
172 of Test=f%(bis%)
173 dx l%=(bis%-von%+1)*VL%
174 u4 l%=(UBOUND(f%)-von%+1)*VL%
175 xQ z8=STRING$(l%,CHR$(0))
176 N9 CALL CopyMem(VARPTR(f%(von%)),SADD(z8),l%)
177 te z8=MID$(z8,l%+1,STRLEN$(l%,CHR$(0)))
178 9Q CALL CopyMem(SADD(z8),VARPTR(f%(von%)),l%)
179 D6 z8=""
180 WY END SUB
181 Ok SUB Search (f%(),ab%,wert%,ps%) STATIC
182 DB VL%=2
183 4h Test=f%(ab%)
184 Sx l%=(UBOUND(f%)+1)*VL%
185 BK pp%=ab%*VL%
186 eH z8=STRING$(l%,CHR$(32))
187 2p CALL CopyMem(VARPTR(f%(0)),SADD(z8),l%)
188 4e z8=""
189 WY WHILE ok=0
190 J2 pp%=VNSR(pp%+1,z8,MKS wert%)
191 rQ ps=pp%/VL%
192 Av z8=VNIS wert% IF ps<0 THEN ok=1
193 YWO WEND
194 A z8=ps
195 5 IF ps=0 THEN ps=-1
196 UN z8=""
197 Dr z8=VNIS wert%
(C) 1989 MET

```

Listing.
Machen Sie mit »Felder« Ihre Basic-Programme schneller. Bitte unbedingt mit dem Checksummer (Seite 143) eingeben.



Es ist soweit:
40 Zaubersprüche war-
ten auf Sie. Eine
Bibliothek voller Funk-
tionen, leichtverständlich
und einfach anzuwenden.
Ob Basic, C oder Assembler, alle Pro-
grammierer haben Vorteile.

ZAUBERN

Vereinfachung von komplizierten und aufwendigen Programmen ist eines der Ziele der Bibliothek »extintui.library«. Dies trifft hauptsächlich für die C-Programmierer zu. Das andere Ziel ist, in Basic Dinge zu realisieren, die bisher nicht möglich waren. Beides ist mit der neuen Bibliothek (Library) gelungen. Der Anwendungsbereich erstreckt sich von Fenstern, über Pull-Down-Menüs mit Unterpunkten bis zur Auswertung von Intuition-Nachrichten. C-Programmierer öffnen wie gewohnt die Bibliothek mit der Funktion `OpenLibrary()`. Das Programm »extintui.lnk« müssen Sie zu Ihrem Hauptprogramm linken. Die Basic-Programmierer haben es einfacher. Sie konvertieren die Datei »extintui.lib.fd« (Listing 3) mit dem Programm »ConvertFD« (Basic-Diskette im Unterverzeichnis BasicDemos). Die erhaltene Datei »extintui.bmap« kopieren Sie zu Ihrem Basic-Programm. Im Listing 4 zeigen wir Ihnen den Vorspann, den Sie am Anfang Ihres Programms verwenden sollten. Damit das System bei eventuellen Aufräumarbeiten alles beseitigen kann, was nutzlos im Speicher steht, muß bei Programmende jede geöffnete Library mit `CloseLibrary()` geschlossen werden. Die Funktionen dienen zur vereinfachten Steuerung der Benutzeroberfläche. Speziell als Anfänger auf dem Amiga ist es schwer, mit den komplexen Strukturen umzugehen, die nötig sind, um ein Fenster zu öffnen oder eine Menüliste erscheinen zu lassen. Die Arbeit wird nun komplett von der Library übernommen. Die Funktion `ShowWindow()` öffnet ein Fenster mit der angegebenen Größe entweder auf dem Workbench-Screen oder einem zuvor geöffneten eigenen Bildschirm. Öffnet man nacheinander ein Fenster, einen Bildschirm und wieder ein Fenster, steht das erste Fenster auf dem Workbench-Screen und das zweite in dem zuvor geöffneten Bildschirm. Wird mit den Zeichenfunktionen der Library ein Kreis gezeichnet, so erscheint dieser immer in dem zuletzt geöffneten Fenster. Ebenso verhält es sich mit den Menü- und Gadget-Funktionen. Zusätzlich bietet die Bibliothek noch einige Fähigkeiten, die bisher von Amiga-Basic aus nicht

Zeichnung: Jan Valach



benutzt werden konnten. So etwa Menüunterpunkte und Requester. Doch zusammen mit anderen Funktionen sind die Wirkungen etwas unterschiedlich. So kann zum Beispiel keine Zeichenfunktion ausgeführt werden, wenn zuvor ein Bildschirm oder Fenster geschlossen wurde. Beim Löschen, werden die Adressen des aktuellen Bildschirms und Fensters, die in einem eigens angefertigten Puffer gespeichert werden, ebenfalls gelöscht. Versucht nun eine Zeichenfunktion, auf dem aktuellen Fenster etwas auszugeben, scheitert die Ausführung, da keine Fenster-Adresse gefunden werden kann. Und nun wollen wir uns den einzelnen Funktionen widmen.

```
screen = ShowScreen(Tiefe,Modus,Titel)
```

SIE MIT

Mit dieser Funktion ist es sehr einfach, einen Screen auf dem Monitor darzustellen. Der Parameter »Tiefe« gibt die Anzahl der Bit-Ebenen und dadurch automatisch die möglichen Farben an. Im Zuge der Vereinfachung wurden für den Parameter »Modus« vier Möglichkeiten festgelegt, die Sie beliebig erweitern oder verändern können:

- 0 = Normalauflösung mit 320 x 256 Pixeln in maximal 64 Farben, was einer Tiefe von 6 entspricht.
- 1 = Hi-Res-Screen mit 640 x 256 Pixeln in maximal 16 Farben und einer Tiefe von maximal 4.
- 2 = Lace-Screen mit 320 x 512 Pixeln in maximal 32 Farben und einer Maximaltiefe von 5.
- 3 = Hi-Res und Lace mit 640 x 512 Pixeln in maximal 16 Farben bei einer Tiefe von 4.

Der Parameter »Titel« enthält den Namen des Bildschirms, der in der Titelleiste erscheint. Die Funktion liefert bei Erfolg die Adresse des Bildschirms zurück. Sie wird benötigt, um den Bildschirm später wieder schließen zu können. Sobald der Screen erscheint, ist er der für das System aktuelle Bildschirm; alle Fensterfunktionen aus der Library beziehen sich auf diesen Screen.

```
SetPal (Farbregister, rot, grün, blau)
```

Mit dieser Funktion setzen Sie die einzelnen Farbregister für Ihren Bildschirm. »Farbregister« ist die Nummer der gewünschten Farbe. Die anderen Parameter können Werte von 0 bis 15 annehmen. 15 bedeutet dabei volle Stärke.

```
ActScreen(screen)
```

Wurden mehrere Screens geöffnet, kann mit dieser Funktion einer ausgewählt werden, der zum aktuellen wird. Der Parameter »screen« ist die Adresse des Bildschirms, der von der Funktion ShowScreen() zurückgeliefert wird.

```
RemScreen(screen)
```

Die Funktion RemScreen() ruft Close Screen() auf, durchläuft zuvor allerdings einige Routinen, die alle auf dem Bildschirm vorhandenen Fenster löscht. Durch diese Funktion werden die Adressen für das aktuelle Fenster und den Bildschirm zurückgesetzt.

```
window = ShowWindow(LeftEdge, TopEdge, Width, Height, IDCMP, Flags, Titel)
```

Zum Erzeugen von Fenstern dient die Funktion ShowWindow(). Sie ist ebenfalls so einfach anzusprechen wie die Funktion ShowScreen(). »LeftEdge« und »TopEdge« geben die linke obere Ecke des Fensters auf dem Bildschirm an. »Width« und »Height« sind die Breite und die Höhe. Die Koordinaten werden überprüft und gegebenenfalls korrigiert. Wird beispielsweise ein Bildschirm in der Größe 320 x 200 Pixel geöffnet, ist es unmöglich, danach ein Fenster in der Größe 600 x 250 Pixel zu öffnen. Nach den Koordinaten werden die IDCMP-Flags für das entsprechende Fenster erwartet. Die IDCMP-Flags, wie auch die Window-Flags, sollen nicht alle aufgelistet werden. Informationen darüber finden Sie in unserem C-Kurs (AMIGA-Magazin, Ausgabe 7/88, Seite 97) und praktisch allen Programmierhandbüchern über den Amiga. »Titel« ist die Adresse eines mit einem Null-Byte abgeschlossenen Strings. Basic-Programmierer müssen daher darauf achten, daß sie ihre Basic-Strings mit einem Null-Byte versehen und die Adresse des Strings an die Funktionen übergeben und nicht etwa den Stringdeskriptor. Zuerst muß die 0 angehängt werden, was folgendermaßen geschieht `a$=a$+CHR$(0)`. Dann verwenden Sie den Befehl SADD, der die Adresse der Zeichenkette berechnet.

```
Act(Window(window))
```

ActWindow() erklärt ein Fenster als aktuell. Aktualisieren bedeutet, daß alle folgenden Zeichenfunktionen auf das in »window« angegebene Fenster umgelenkt werden. Das Fenster wird nicht in den Vordergrund gestellt.

```
RemWindow(window)
```

Diese Funktion dient zum Löschen einzelner Fenster. Sollte das Fenster das aktuelle sein, wird die Adresse innerhalb des speziellen Datenbereichs auf Null gesetzt. Zeichenfunktionen können dann erst wieder nach einem ShowWindow()- oder ActWindow()-Aufruf ausgeführt werden. Die Zeichen-, Textausgabe- und Nachfragefunktionen der Library beziehen sich, wie bereits erwähnt, auf das jeweils aktuelle Fenster. Deshalb wird bei keiner dieser Funktionen die Fenster- oder RastPort-Adresse verlangt. Sie müssen lediglich darauf achten, daß ein aktuelles Fenster vorhanden ist. Sollen Ausgaben in verschiedenen, bereits geöffneten Fenstern stattfinden, müssen die jeweiligen Fenster vor der Ausgabe mit ActWindow() aktualisiert werden.

```
PutCircle(x,y,radius)
```

```
PutEllipse(x,y,vradius,hradius)
```

PutCircle() zeichnet einen Kreis und PutEllipse() eine Ellipse in das aktuelle Fenster. »x« und »y« sind dabei jeweils der Mittelpunkt. »vradius« ist der vertikale und »hradius« der horizontale Radius der Ellipse. Bei einem Kreis sind beide Werte gleich, daher muß nur der Radius im Parameter »radius« angegeben werden.

```
PutPoint(x,y)
```

Setzt an der Position mit den Koordinaten »x« und »y« einen Punkt in der aktuellen Farbe.

```
PutLine(x1,y1,x2,y2)
```

PutLine() zieht eine Linie vom Punkt »x1« und »y1« zu den Koordinaten »x2« und »y2«. Es wird die aktuelle Farbe verwendet.

```
PrintText(Text,x,y)
```

Zur Grafiktextausgabe wird die Funktion PrintText() eingesetzt. Der Text wird dabei in der aktuellen Zeichenfarbe und dem Modus ausgegeben. Die Position wird wieder mit »x« und »y« festgelegt.

```
SetColor(front,back,out)
```

```
Mode(Drawmode)
```

Die Funktionen SetColor() und Mode() schreiben die angegebenen Werte für Farben und Zeichenmodi direkt in den RastPort des aktuellen Fensters. Wechseln Sie mit ActWindow() das aktuelle Fenster, müssen die Farb- und Zeichenmodi neu eingestellt werden, wenn dies nicht bereits geschah. »front« und »back« sind die Vorder- und Hintergrundfarbe. Mit »out« geben Sie die Farbe für Umrahmungen an.

```
x=GetPosX()
```

```
y=GetPosY()
```


Die GetPos()-Funktionen ermitteln die aktuelle Position des Grafik-Cursors innerhalb des aktuellen Fensters. Basic-Programmierer müssen hier wie bei den anderen Funktionen, die keinen Parameter erwarten, eine »0« einsetzen, da sonst Amiga-Basic einen Syntax-Fehler anzeigt. Der Grafik-Cursor wird beim Zeichnen verwendet. Es werden zwei getrennte Funktionen verwendet, da sonst von Basic aus wieder mit Zeigern gearbeitet werden müßte. Dasselbe gilt für die folgenden zwei Funktionen:

```
x=MouseX()      x=MouseY()
```

MouseX() und MouseY() geben Auskunft über die Position des Mauszeigers innerhalb des Fensters. Das Erzeugen von Amiga-Menüleisten ist in C eine komplizierte Sache. Es müssen Strukturen versorgt und miteinander verbunden werden. Die Menüfunktionen der Library machen dies automatisch und müssen nur in der Reihenfolge, in der das Menü angezeigt werden soll, aufgerufen werden.

Die Größe des Selektierungsbereichs der einzelnen Menüeinträge richtet sich nach dem längsten im gesamten Menü enthaltenen Text. Somit ist gewährleistet, daß alle Menüeinträge in der gleichen Größe erscheinen. Alle Menüfunktionen mit Ausnahme von KillMenu() beziehen sich auf das aktuelle Fenster. Ist kein Fenster vorhanden, auf dem eine Menüleiste angezeigt werden kann, werden die Funktionen erfolglos abgebrochen.

```
MenuHeader(text)
```

erzeugt eine neue Menüleiste. Ist bereits eine Menüleiste für das aktuelle Fenster generiert worden, wird die neue Menüleiste rechts daneben angezeigt.

```
MenuItem(text)
```

stellt einen neuen Menüeintrag unter eine Menüleiste oder einen bereits vorhandenen Menüeintrag. Ein Menüeintrag kann nur erzeugt werden, wenn eine Menüleiste vorhanden ist.

```
MenuSubItem(text)
```

Untereinträge werden immer an zuletzt erzeugte Untereinträge oder Menüeinträge angehängt. Diese Funktion kann nicht ausgeführt werden, wenn die Menüleiste keinen Menüeintrag enthält oder keine Menüleiste vorhanden ist.

```
KillMenu(window)
```

Hat eine Menüleiste ihren Dienst getan, muß die Funktion KillMenu() aufgerufen werden. Der von der Menüleiste belegte Speicherplatz wird dann dem System wieder zur Verfügung gestellt. Die Menüleiste ist nicht mehr anwählbar. Unabhängig vom aktuellen Fenster muß die Fenster-Adresse mitgeteilt werden.

Fenster, Gadgets und Requester

Komplizierter als Menüleisten sind Gadgets und Requester. Wo bei Menüleisten nur zwei Strukturen versorgt werden müssen, sind es bei Gadgets gleich vier. Dies liegt an der vielseitigen Gestaltungs- und Anwendungsmöglichkeit von Gadgets. Bei Boolean-Gadgets ist dies vielleicht noch recht einfach, bei Proportional- oder String-Gadgets müssen zusätzlich noch Info-Strukturen angelegt werden. In der Library besorgen dies die Funktionen SetPropInfo() und SetStringInfo(), die das Ergebnis »Info« liefern. Der Parameter »Info« wird dann der SetGadget()-Funktion übergeben. Genau wie bei den Menüleisten, müssen Sie je nach gewünschter Anzahl der Gadgets die Funktion SetGadget() aufrufen. Erst nach Öffnen eines Requesters mit der Funktion ShowRequester() werden die zuvor mit SetGadget() erzeugten Gadgets im aktuellen Fenster angezeigt. Die Reihenfolge, in der die Funktionen aufgerufen werden müssen, lautet wie folgt:

1. Fenster öffnen mit ShowWindow()
2. Gadgets erzeugen mit SetGadget()
3. Requester öffnen mit ShowRequester()

```
strinfo = SetStringInfo(maxchars)
```

Reserviert Speicherplatz für eine StringInfo-Struktur und liefert deren Adresse als Ergebnis. Der Parameter »maxchars« legt die Anzahl der Zeichen fest, die in das Stringfeld eingegeben werden können. Die zurückerhaltene Adresse muß der Funktion SetGadget() im Parameter »Info« mitgeteilt werden. Außerdem ist der Para-

meter »Typ« der Funktion SetGadget() mit dem Wert der Definition »STRGADGET« (4) zu versorgen.

```
propinfo = SetPropInfo(Flags,hpot,vpot,hbody,vbody)
```

Die Funktion SetPropInfo reserviert Speicherplatz für eine PropInfo-Struktur. Die zurückgelieferte Strukturadresse muß, wie bei der Funktion SetStringInfo(), an den Parameter »Info« der Funktion SetGadget() übergeben werden. Im Parameter »Flags« wird mitge-

Gadgets am richtigen Platz

teilt, ob sich der Slider in vertikaler (4), horizontaler (2) oder in beiden Richtungen (6) bewegen läßt. Die Parameter »hpot« und »vpot« bestimmen die Position des Reglers innerhalb des Gadgets. Genau wie »hbody« und »vbody«, können die Parameter »hpot« und »vpot« Werte von 0 (klein) bis 65335 (groß) annehmen.

```
gadget = SetGadget(LeftEdge,TopEdge,Width,Height,Activ,Typ,Text,Info,Id)
```

Mit dieser Funktion werden die Gadgets erzeugt, aber noch nicht auf dem Bildschirm angezeigt. Die Größe und Position des Gadgets innerhalb des Requesters, wird mit den Parametern »LeftEdge«, »TopEdge«, »Width« und »Height« festgelegt. Sie haben dieselbe Bedeutung wie bei ShowWindow(). In »Activ« wird angegeben, welche Nachricht das Gadget liefern soll, wenn es angeklickt wird, oder speziell bei String-Gadgets, wie der eingegebene Text im Stringfeld erscheinen soll.

```
RELVERIFY ( 1 ) > GADGETUP
```

```
GADGIMMEDIATE ( 2 ) > GADGETDOWN
```

Diese beiden »Activ«-Werte müssen gesetzt sein, wenn das Gadget eine IntuiMessage erzeugen soll, sobald es mit dem Mauszeiger angeklickt wird. Das Fenster, in dem der Requester mit den Gadgets angezeigt wird, muß dazu in den IDCMP-Flags die Werte für GADGETDOWN oder GADGETUP enthalten.

Für String-Gadgets können zusätzlich noch folgende Werte gesetzt werden:

- STRCENTER (512) : Die Eingabe erfolgt zentriert.
 - STRRIGHT (1024) : Die Eingabe erfolgt rechtsbündig.
 - LONGINT (2048) : Es können nur Ziffern eingegeben werden.
- Der Parameter »Typ« muß Informationen über die Art des Gadgets enthalten. Bei einem Boolean-Gadget den Wert 1 für »BOOL GADGET«. Der Parameter »Info« wird für Boolean-Gadgets nicht berücksichtigt. Für String-Gadgets muß der Parameter »Typ« den Wert 4 für »STRGADGET«, und der Parameter »Info« die Adresse der reservierten StringInfo-Struktur, enthalten. Bei Proportional-Gadgets wird in »Typ« der Wert 3 für »PROPGADGET« und in »Info« die Adresse einer PropInfo-Struktur erwartet.

»Text« ist ein mit einem Null-Byte beendeter String, der im Gadget angezeigt wird. In Boolean-Gadgets wird der Text innerhalb des Gadgets zentriert ausgegeben. Bei Proportional-Gadgets steht der Text über dem Gadget, da der Slider ihn überdecken würde. Bei String-Gadgets steht er links neben dem Eingabefeld.

Die Kennziffer oder Id des Gadgets wird im Parameter »id« erwartet. Die Gadget-Kennung ist wichtig, um bei einer aufgetretenen Nachricht zu erfahren, welches Gadget angeklickt wurde. Tritt eine GADGETDOWN- oder GADGETUP-Nachricht ein, kann mit der Funktion Gadgetid() die Kennung des angeklickten Gadgets erfragt und entsprechend reagiert werden.

```
textadr = GetStringInfo(requester,id)
```

Erhält man die GADGETUP-Nachricht von einem String-Gadget, liefert die Funktion GetStringInfo() die Adresse des eingegebenen Strings zurück. Der Parameter »requester« wird von der Funktion ShowRequester() geliefert und muß zur Identifizierung des Gadgets angegeben werden. »id« muß die Kennung des Gadgets enthalten.

Die nachfolgenden Funktionen benötigen, gleich der Funktion GetStringInfo(), die Parameter »requester« und »id« zur genauen Lokalisierung des Gadgets.

```
x = GetPropX(requester,id)
```

```
y = GetPropY(requester,id)
```

GetPropX() und GetPropY() liefern die Position des Reglers innerhalb eines Proportional-Gadgets. Achten Sie bitte darauf, daß Sie die Variablen für den Empfang der Werte so definieren, daß der Wertebereich von 0 bis 65335 reicht. Basic-Programmierer ver-

knüpfen das Ergebnis in diesem Fall logisch mit 65535 (x% AND 65535&).

```
w = GetPropW(requester,id)
h = GetPropH(requester,id)
```

Die beiden Funktionen liefern die Größe des Reglers im Bereich 0 bis 65535.

```
ChangeProp(requester,id,hpot,vpot,hbody,vbody)
```

Damit die Größe und Position des Reglers innerhalb eines Proportional-Gadgets geändert werden kann, steht die Funktion ChangeProp() zur Verfügung. Zur Requester-Adresse und Gadget-Kennung müssen die neuen Ausmaße des Sliders angegeben werden.

```
requester = ShowRequester(LeftEdge, TopEdge,
Width, Height, Farbe)
```

ShowRequester() öffnet einen Requester in der angegebenen Größe und Farbe im aktuellen Fenster. Wurden zuvor Gadgets erzeugt, werden diese im Requester mit angezeigt. Hatte die Funktion Erfolg, wird die Adresse des Requesters zurückgeliefert. Sie wird benötigt, um den Requester wieder zu schließen. Kann kein Requester geöffnet werden, wird eine Null zurückgeliefert.

```
RequesterEnd(requester)
```

Die Funktion RequesterEnd() läßt den Requester wieder verschwinden und gibt den durch die Gadgets belegten Speicherplatz frei. Soll der gleiche Requester erneut geöffnet werden, müssen alle darin enthaltenen Gadgets neu erzeugt werden.

Der Nachrichtenempfang bezieht sich immer auf ein Fenster. Beim Öffnen eines Fensters muß deshalb in den IDCMP-Flags mitgeteilt werden, welche Nachrichten das jeweilige Fenster empfangen kann. Wird beispielsweise der Wert für MOUSEBUTTONS an das IDCMP-Flag des Fensters übergeben, erhält man immer eine Nachricht, wenn eine der Maustasten gedrückt wird.

```
ok = Message(window)
```

Message() liefert 1, wenn für das angegebene Fenster eine Nachricht aufgetreten ist, andernfalls eine 0. Ist eine Nachricht eingegangen, kann mit der Funktion Class() die Art (IDCMP) der Nachricht erfragt werden. Die Funktion sollte in einer Schleife aufgerufen werden, die erst dann wieder verlassen wird, wenn ein bestimmtes Ereignis eintritt.

```
class = Class()
```

Class() liefert den Class-Wert der letzten Nachricht.

```
code = Code()
```

Code() gibt den Code-Wert der letzten Nachricht zurück.

```
id = GadgetId()
```

• Enthält der Class-Wert der eingetretenen Nachricht den Wert

GADGETDOWN oder GADGETUP, ermittelt die Funktion GadgetId() die Kennung des angeklickten Gadgets.

Ist die Nachricht MENUICK eingetreten, kann mit den nachfolgenden Funktionen der angewählte Menüeintrag oder Untereintrag ermittelt werden:

```
id = Menuid()
```

Liefert die Kennung der angewählten Menüleiste.

```
id = Itemid()
```

Liefert die Kennung des angewählten Menüeintrags.

```
id = Subid()
```

Liefert die Kennung des angewählten Untermenüeintrags.

Die Routinen für Funktionen

Das Prinzip, in Form von Bibliotheken (Libraries) Routinen zur Programmierung bereitzustellen, gehört zum Amiga wie das Salz in die Suppe. Um den großen Aufwand der Vorbereitung und Verwaltung zu verkleinern, gibt es jetzt unsere Bibliothek »extintui.library«. Kurze Programme mit großer Wirkung sind kein Wunschtraum mehr.

Bisher fanden Sie die genaue Beschreibung der einzelnen Funktionen. Was noch fehlt, sind die Routinen selbst. Die finden Sie in Listing 5. Zusammen mit den Listings aus der letzten Ausgabe können Sie nun die Bibliothek generieren. Falls Sie den benötigten Aztec-C-Compiler nicht besitzen, finden Sie alle Programme in lauffähiger Version auf der Programmservice-Diskette.

Für Basic-, C- und Assembler-Programmierer war bis jetzt der Zugriff auf die 40 Funktionen der neuen Bibliothek »extintui.library« möglich. Damit auch die Modula-2-Fans in den Genuß der einfachen Programmierung kommen, finden Sie im Listing 6 das nötige Modula-2-Programm. Compilieren Sie es und kopieren Sie die generierte Datei »M2Extintui.sym« in das Verzeichnis »Modules/Interfaces/sym« Ihrer Modula-2-Systemdiskette.

Die formalen Parameter der Prozeduren sind gemäß der in »Intuition.def« und »Graphics.def« festgelegten Typendeklarationen benannt. Dadurch wird der eventuelle Umstieg in die direkte Programmierung von Intuition erleichtert. Das Umsetzen des Demoprogramms dürfte ein Kinderspiel sein.

Und nun Start frei für kurze, benutzerfreundliche Programme in vier Programmiersprachen. *Jürgen Haage/Andreas Görtler/kn*

Compiler- und Assembleranweisungen für Listing 1:

```
as -o extintui__lib.o extintui__lib.asm -D -N
```

```
as -o extintui__lnk.o extintui__lnk.asm -D -N
```

Programmname: extintui__lib.asm			
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3		
Sprache:	Assembler		
Bemerkung:	siehe Text		
Programmautor: Jürgen Haage			

1	7b0 *//*****	10	th include "exec/initialisera.1"
2	wq /* extintui__lib.o v1.0	11	kf include "exec/libraries.1"
3	lj /* Jürgen Haage Burgstraße 30	12	fp include "exec/lists.1"
4	yt /* 6552 Bad Münster a. St. Ebernbr	13	lu include "exec/resident.1"
5	bt /******	14	vm include "exec/strings.1"
6	cd VERSION equ 1	15	lc include "libraries/dos.1"
7	kc REVISION equ 1	16	jd include "exec/lo.1"
8	lw3 nullist	17	ou list
9	rp include "exec/types.1"	18	ke STRUCTURE ExtintuiBase,LIB_SI
			ZE
		19	fn RPTR extin_SegList
		20	ao APTR extin_SysBase
		21	eo APTR extin_IntuitionBase
		22	yvs APTR extin_GfxBase
		23	bb3 LABEL extin_SIZEOF
		24	exd EXTINTUILIBNAME macro
		25	bx3 dc.b 'extintui.library',0
		26	lj3 endm
		27	7b0 xlib macro
		28	303 xref _LVO%1
		29	4a endm
		30	xY0 callsys macro
		31	9p3 jsr _LVO%1(a6)
		32	7p endm
		33	YR xref FindLib
		34	yd xref ShowWindow
		35	bY xref ShowScreen
		36	UD xref _ActScreen
		37	q4 xref _ResScreen
		38	Re xref _ActWindow
		39	nQ xref _ResWindow
		40	1B xref _PutCircle
		41	%6 xref _PutEllipse
		42	Gb xref _PutLine
		43	2E xref _PutPoint
		44	OP xref _SetColor
		45	VL xref _Mode
		46	m5 xref _GetPosX
		47	14 xref _GetPosY
		48	aj xref _PrintText
		49	1Y xref _MouseX
		50	x5 xref _MouseY
		51	ae xref _MenuHeader
		52	K2 xref _MenuItem
		53	IX xref _MenuSubItem
		54	Ga xref _KillMenu
		55	aj xref _SetStringInfo
		56	5L xref _SetPropInfo
		57	7W xref _SetGadget
		58	0x xref _GetStringInfo

Listing 1. »extintui__lib.asm« ist der Rumpf der Bibliothek

```

59 xref _GetPropX
60 xref GetPropY
61 xref GetPropW
62 xref _GetPropH
63 xref _ChangeProp
64 xref _ShowRequester
65 xref _RequesterEnd
66 xref _Message
67 xref Class
68 xref Code
69 xref Gadgetid
70 xref Menuid
71 xref Itemid
72 xref Subid
73 xref SetPal
74 xlib OpenLibrary
75 xlib CloseLibrary
76 xlib FindTask
77 xlib Remove
78 xlib FreeMem
79 xdef _SysBase
80 xdef IntuitionBase
81 xdef GfxBase
82 xdef _AllocMem
83 CSEG
84 ar moveq #RETURN_FAIL,d0
85 rts
86 ROMTag:
87 dc.w RTC_MATCHWORD
88 dc.l ROMTag
89 dc.l EndCode
90 dc.b RTE_AUTOINTT
91 dc.b VERSION
92 dc.b NL_LIBRARY
93 dc.b 0
94 dc.l libraryName
95 dc.l idString
96 dc.l Init
97 oword libraryName:
98 EXTINTUILLIBNAME
99 idString:
100 dc.b 'extintuilib 1.0 (12.OKT
1988)',CR,LF,0
101 intuilib:
102 dc.b 'intuition.library',0
103 ;cnop 0,2
104 gfxlib:
105 dc.b 'graphics.library',0
106 ;cnop 0,2
107 Init:
108 dc.l extin_SIZEOF
109 dc.l funcTable
110 dc.l dataTable
111 dc.l .InitLib
112 vword funcTable:
113 dc.l .OpenLib
114 dc.l .CloseLib
115 dc.l .ExpungeLib
116 dc.l .ExtFuncLib
117 dc.l .FindLib
118 dc.l .ShowWindow
119 dc.l .ShowScreen
120 dc.l .ActScreen
121 dc.l .RemScreen
122 dc.l .ActWindow
123 dc.l .RemWindow
124 dc.l .PutCircle
125 dc.l .PutEllipse
126 dc.l .PutLine
127 dc.l .PutPoint
128 dc.l .SetColor
129 dc.l .Mode
130 dc.l .GetPosX
131 dc.l .GetPosY
132 dc.l .PrintText
133 dc.l .MouseX
134 dc.l .MouseY
135 dc.l .MenuHeader
136 dc.l .MenuItem
137 dc.l .MenuSubItem
138 dc.l .KillMenu
139 dc.l .GetStringInfo
140 dc.l .SetPropInfo
141 dc.l .SetGadget
142 dc.l .GetStrInfo
143 dc.l .GetPropX
144 dc.l .GetPropY
145 dc.l .GetPropW
146 dc.l .GetPropH
147 dc.l .ChangeProp
148 dc.l .ShowRequester
149 dc.l .RequesterEnd
150 dc.l .Message
151 dc.l .Class
152 dc.l .Code
153 dc.l .Gadgetid
154 dc.l .Menuid
155 dc.l .Itemid
156 dc.l .Subid
157 dc.l .SetPal
158 dc.l -1
159 sq0 dataTable:
160 sq0 INITBYTE LH_TYPE,NL_LIBRARY
161 sq0 INITLONG LN_NAME,libraryName
162 sq0 INITBYTE LIB_FLAGS,LIBF_SUMUS
ED:LIBF_CHANGED
163 sq0 INITWORD LIB_VERSION,VERSION
164 sq0 INITWORD LIB_REVISION,REVISIO
N
165 sq0 INITLONG LIB_IDSTRING,idStrin
g
166 dc.l 0
167 InitLib:
168 move.l a5,-(sp)
169 movea.l d0,a5
170 move.l a0,extin_SegList(a5)
171 lea intuilib(pc),a1
172 moveq #LIBRARY_VERSION,d0
173 callsys OpenLibrary
174 tst.l d0
175 beq.s 1$
176 move.l a6,extin_SysBase(a5)
177 move.l a6,_SysBase
178 move.l d0,extin_IntuitionBase
(a5)
179 move.l d0,_IntuitionBase
180 lea gfxlib(pc),a1
181 moveq #LIBRARY_VERSION,d0
182 callsys OpenLibrary
183 tst.l d0
184 beq 1$
185 move.l d0,extin_GfxBase(
a5)
186 move.l d0,OfxBase
187 move.l a5,d0
188 movea.l (sp)+,a5
189 rts
190 .OpenLib:
191 addq.w #1,LIB_OPENCNT(a6)
192 bclr #LIBB_DELEXP,LIB_FLAGS(
a6)
193 move.l a6,d0
194 movea.l d0/a1,a5,-(sp)
195 move.l a6,a5
196 move.l 16(sp),a6
197 pea 65536 MEMM_CLEAR
198 pea 100
199 jsr _AllocMem Speicher anf
ordern
200 add.w #8,sp
201 tst.l d0
202 beq 1$
203 move.l d0,-(sp)
204 move.l #0,a1
205 callsys FindTask Taskadress
e ermitteln
206 move.l d0,a1
207 move.l (sp)+,d0
208 move.l d0,88(a1) Adresse de
s Speicherbereiches
209 move.l a5,a6
210 movea.l (sp)+,d0/a1/a5
211 rts
212 sq0 1$
213 movea.l (sp)+,d0/a1/a5
214 move.l #0,d0
215 rts
216 rts
217 rts
218 CloseLib:
219 movea.l a1-a2/#%,-(sp)
220 move.l a6,a5
221 move.l extin_SysBase(a5),a6
222 sub.l a1,a1
223 callsys FindTask
224 tst.l d0
225 beq.s 00$
226 move.l d0,a2 ;Taskadresse erm
itteln
227 move.l 88 a2,d0
228 beq.s 01$
229 move.l d0,a1
230 move.l #100,d0
231 callsys FreeMem ;user Speic
her wieder freigeben
232 movea.l (sp)+,a1-a2/a5
233 move.l a5,a6
234 subq.w #0,d0
235 subq.w #1,LIB_OPENCNT(a6)
236 bne.s 1$
237 btst #LIBB_DELEXP,LIB_FLAGS(
a6)
238 beq.s 1$
239 bsr.s .ExpungeLib
240 rts
241 .ExpungeLib:
242 movea.l d2/a5/a6,-(sp)
243 tst.w LIB_OPENCNT(a6)
244 beq.s 1$
245 bset #LIBB_DELEXP,L.B_FLAGS
(a6)
246 moveq #0,d0
247 bra.s 2$
248 move.l a6,a5
249 move.l extin_SysBase(a5),a6
250 movea.l extin_IntuitionBase(a
5),a1
251 callsys CloseLibrary
252 movea.l extin_GfxBase(a5),a1
253 callsys CloseLibrary
254 move.l a5,a1
255 callsys Remove
256 move.l extin_SegList(a5),d2
257 movea.l a5,a1
258 moveq #0,d0
259 move.w LIB_NEGSIZE(a5),d0
260 suba.l d0,a1
261 add.w LIB_POSSIZE(a5),d0
262 callsys FreeMem
263 move.l d2,d0
264 movea.l (sp)+,d2/a5/a6
265 movea.l d2/a5/a6
266 rts
267 .ExtFuncLib:
268 moveq #0,d0
269 rts
270 FindLib:
271 jsr _FindLib(pc)
272 rts
273 ShowWindow
274 move.l a0,-(a7)
275 movea.l d0-d5,-(a7)
276 jsr _ShowWindow(pc)
277 adda.w #28,sp
278 rts
279 ShowScreen
280 move.l a0,-(a7)
281 movea.l d0-d1,-(a7)
282 jsr _ShowScreen(pc)
283 adda.w #12,sp
284 rts
285 ActScreen
286 move.l a0,-(a7)
287 jsr _ActScreen(pc)
288 addq.l #4,sp
289 rts
290 RemScreen
291 move.l a0,-(a7)
292 jsr _RemScreen(pc)
293 addq.l #4,sp
294 rts
295 ActWindow

```



```

296 lb3 move.l a0,-(a7)
297 ld jsr _ActWindow(pc)
298 fd addq.l #4,sp
299 fd rts
300 Q20 RemWindow
301 Qg3 move.l a0,-(a7)
302 xQ jsr _RemWindow(pc)
303 K9 addq.l #4,sp
304 fz rts
305 noQ .PutCircle
306 A3 move.l d0-d2,-(a7)
307 Af jsr _PutCircle(pc)
308 Ab adda.w #12,sp
309 fd rts
310 A8Q .PutEllipse
311 Po3 move.l d0-d3,-(a7)
312 M jsr _PutEllipse(pc)
313 Lk adda.w #16,sp
314 B9 rts
315 x0Q .PutLine
316 Ab3 move.l d0-d3,-(a7)
317 fw jsr _PutLine(pc)
318 G1 adda.w #16,sp
319 B5 rts
320 Q0 .PutPoint
321 P43 move.l d0-d1,-(a7)
322 r0 jsr _PutPoint(pc)
323 M1 addq.l #8,sp
324 L rts
325 x4Q .SetColor
326 Ap3 move.l d0-d2,-(a7)
327 LQ jsr _SetColor(pc)
328 O4 adda.w #12,sp
329 M0 rts
330 D5Q .Mode
331 CG3 move.l d0,-(a7)
332 Gw jsr _Mode(pc)
333 od addq.l #4,sp
334 ST rts
335 nXQ .GetPosX
336 nA3 jsr _GetPosX(pc)
337 w rts
338 w0 .GetPosY
339 VQ3 jsr _GetPosY(pc)
340 z rts
341 w0 .PrintText
342 K33 move.l d0-d1,-(a7)
343 M move.l a0,-(a7)
344 z jsr _PrintText(pc)
345 f0 adda.w #12,sp
346 w rts
347 9PQ .MouseX
348 TP3 jsr _MouseX(pc)
349 r1 rts
350 FwQ .MouseY
351 xK3 jsr _MouseY(pc)
352 k1 rts
353 x0 .MenuHeader
354 X3 move.l a0,-(a7)
355 M jsr _MenuHeader(pc)
356 D0 addq.l #4,sp
357 f4 rts
358 dQ .MenuItem
359 Ab3 move.l a0,-(a7)
360 J jsr _MenuItem(pc)
361 G5 addq.l #4,sp
362 w rts
363 Y0 .MenuSubItem
364 Rh3 move.l a0,-(a7)
365 dC jsr _MenuSubItem(pc)
366 LA addq.l #4,sp
367 z0 rts
368 LEQ .KillMenu
369 m3 move.l a0,-(a7)
370 P jsr _KillMenu(pc)
371 QF addq.l #4,sp
372 w rts
373 d1Q .SetStringInfo
374 B33 move.l a0,-(a7)
375 z1 jsr _SetStringInfo(pc)
376 Vx addq.l #4,sp
377 9A rts
378 SxQ .SetPropInfo
379 dx3 move.l d0-d4,-(a7)

```

```

380 a8 jsr _SetPropInfo(pc)
381 B8 adda.w #20,sp
382 FF rts
383 xQ .SetGadget
384 A33 move.l a0-a2,-(a7)
385 Ps move.l d0-d5,-(a7)
386 G4 jsr _SetGadget(pc)
387 oE adda.w #36,sp
388 k1 rts
389 rQ .GetStringInfo
390 Q33 move.l d0,-(a7)
391 B5 move.l a0,-(a7)
392 X1 jsr _GetStringInfo(pc)
393 L5 addq.l #8,sp
394 G1 rts
395 xQ .GetPropX
396 Q33 move.l d0,-(a7)
397 X3 move.l a0,-(a7)
398 V4 jsr _GetPropX(pc)
399 t addq.l #4,sp
400 wX rts
401 wQ .GetPropY
402 P33 move.l d0,-(a7)
403 K4 move.l a0,-(a7)
404 L3 jsr _GetPropY(pc)
405 z2 addq.l #8,sp
406 d rts
407 rQ .GetPropH
408 V33 move.l d0,-(a7)
409 J4 move.l a0,-(a7)
410 rC jsr _GetPropH(pc)
411 B5 addq.l #8,sp
412 L rts
413 Q0 .GetPropW
414 Ab3 move.l d0,-(a7)
415 J4 move.l a0,-(a7)
416 rC jsr _GetPropW(pc)
417 B5 addq.l #8,sp
418 L rts
419 Q0 .ChangeProp
420 P33 move.l d0-d4,-(a7)
421 K4 move.l a0,-(a7)
422 X4 jsr _ChangeProp(pc)
423 J3 adda.w #24,sp
424 wV rts
425 e0 .ShowRequester
426 V33 move.l d0-d4,-(a7)
427 9t jsr _ShowRequester(pc)
428 w3 adda.w #20,sp
429 zG rts
430 F3Q .RequesterEnd
431 Ab3 move.l a0,-(a7)
432 V4 jsr _RequesterEnd(pc)
433 LF addq.l #4,sp
434 w5 rts
435 xQ .Message
436 Br3 move.l a0,-(a7)
437 K4 jsr _Message(pc)
438 t addq.l #4,sp
439 M rts
440 xQ .Class
441 VY3 jsr _Class(pc)
442 L rts
443 Q0 .Code
444 A33 jsr _Code(pc)
445 r3 rts
446 r0 .GadgetId
447 C3 jsr _GadgetId(pc)
448 L3 rts
449 B0 .MenuId
450 P33 jsr _MenuId(pc)
451 L3 rts
452 o1Q .Itemid
453 A23 jsr _Itemid(pc)
454 J3 rts
455 tQ .Subid
456 L33 jsr _Subid(pc)
457 9S rts
458 xE0 .SetPal
459 Q33 move.l d0-d3,-(a7)
460 JW jsr _SetPal(pc)
461 Jx adda.w #16,sp
462 wX rts
463 VYD EndCode:

```

```

464 B#3 .BA* by base,4
465 L7 .L7* by IntuitionBase,4
466 90 GLOBAL _GfxBase,4
467 v0 end
(c) 1988 M&T

```

Listing 1. (Schluß)

Programmname:	extintui__lnk.asm
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Assembler
Bemerkung:	siehe Text

Programmautor: Jürgen Haage

```

1 *00 * *****
2 IR /* extintui__lnk.c v1.0
3 IJ /* Jürgen Haage Burgstraße 30
4 VT /* 6552 Bad Münst. a. St. Ebernbg
5 /* *****
6 BM3 XDEF _ExtintuiBase
7 d0 XDEF _ShowWindow
8 Gv XDEF _ShowScreen
9 92 XDEF _ActScreen
10 Vx XDEF _RemScreen
11 6z XDEF _ActWindow
12 J4 XDEF _RemWindow
13 No XDEF _PutCircle
14 2T XDEF _PutEllipse
15 vy XDEF _PutLine
16 r0 XDEF _SetColor
17 Jm XDEF _Mode
18 A1 XDEF _GetPosX
19 RS XDEF _GetPosY
20 YV XDEF _PrintText
21 Xg XDEF _MouseX
22 QU XDEF _MouseY
23 Wb XDEF _MenuItem
24 eY XDEF _MenuItem
25 tr XDEF _MenuItem
26 Ov XDEF _KillMenu
27 VF XDEF _SetStringInfo
28 n6 XDEF _SetPropInfo
29 k1 XDEF _SetGadget
30 ms XDEF _GetStringInfo
31 fN XDEF _GetPropX
32 e0 XDEF _GetPropY
33 h4 XDEF _GetPropH
34 z2 XDEF _GetPropW
35 J4 XDEF _ChangeProp
36 J1 XDEF _ShowRequester
37 fy XDEF _RequesterEnd
38 gW XDEF _Message
39 Ih XDEF _Class
40 HS XDEF _Code
41 J7 XDEF _GadgetId
42 X1 XDEF _MenuId
43 km XDEF _Itemid
44 Wm XDEF _Subid
45 Bz XDEF _SetPal
46 zR XDEF _SetPal
47 tV XDEF _SetPal
48 F3Q ShowWindow
49 Yq3 move.l a6,-(a7)
50 LK move.l _ExtintuiBase,a6
51 vU move.l 8(a7),d0-d5

```

Listing 2.
C-Programmierer linken »extintui__lnk.asm« zu ihrem Programm

54 ME	move.l 32(a7),a0	136 97	move.l _ExtIntuiBase,a6	220 DW	jsr -180(a6)
55 NS	jsr -36(a6)	137 98	jsr -176(a6)	221 20	movea.l (a7)+,a6
56 LC	movea.l (a7)+,a6	138 99	movea.l a7+,a6	222 21	rtss
57 xy	rtss	139 100	rtss	223 3B0	GetPropX.
58 2G0	ShowScreen	140 101	movea.l _ExtIntuiBase,a6	224 HJ3	move.l a6,-(a7)
59 g23	move.l a6,-(a7)	141 102	movea.l _ExtIntuiBase,a6	225 2M	movea.l _ExtIntuiBase,a6
60 tf	movea.l _ExtIntuiBase,a6	142 103	movea.l a7+,a6	226 7	move.l 12(a7),d0
61 vQ	move.l 8(a7),d0-d1	143 104	jsr -114(a6)	227 7	movea.l 8(a7),a0
62 QU	move.l 16(a7),a0	144 105	movea.l a7+,a6	228 1h	jsr 186(a6)
63 2N	jsr -42(a6)	145 106	rtss	229 AI	movea.l (a7)+,a6
64 TK	movea.l (a7)+,a6	146 107	rtss	230 2N	rtss
65 56	rtss	147 108	move.l a6,-(a7)	231 GP0	GetPropY.
66 2W0	_ActScreen	148 109	movea.l _ExtIntuiBase,a6	232 V23	move.l a6,-(a7)
67 oA3	move.l a6,-(a7)	149 110	move.l 8(a7),a0	233 1U	movea.l _ExtIntuiBase,a6
68 1n	movea.l _ExtIntuiBase,a6	150 111	movea.l 12(a7),d0-d1	234 FI	move.l 12(a7),d0
69 uI	movea.l 8(a7),a0	151 112	jsr -120(a6)	235 2D	movea.l 8(a7),a0
70 8r	jsr -48(a6)	152 113	movea.l (a7)+,a6	236 Xn	jsr -192(a6)
71 2D	rtss	153 114	rtss	237 19	movea.l (a7)+,a6
72 2d0	_RemScreen.	154 115	MouseX:	238 2V	rtss
73 vH3	move.l a6,-(a7)	155 116	move.l a6,-a7	239 210	_GetPropW:
74 8.	movea.l _ExtIntuiBase,a6	156 117	movea.l _ExtIntuiBase,a6	240 2a3	move.l a6,-(a7)
75 2P	move.l 8(a7),a0	157 118	jsr -126(a6)	241 2e	movea.l _ExtIntuiBase,a6
76 2L	jsr -54(a6)	158 119	movea.l (a7)+,a6	242 NQ	move.l 12(a7),d0
77 nf	movea.l (a7)+,a6	159 120	rtss	243 78	movea.l 8(a7),a0
78 1K3	rtss	160 121	rtss	244 17	jsr -198(a6)
79 210	_ActWindow.	161 122	movea.l a6, a	245 2H	movea.l a7+,a6
80 203	move.l a6,-(a7)	162 123	movea.l _ExtIntuiBase,a6	246 23	rtss
81 FI	movea.l _ExtIntuiBase,a6	163 124	jsr -132(a6)	247 910	_GetPropH
82 V*	movea.l 8(a7),a0	164 125	movea.l a7+,a6	248 173	move.l a6,-(a7)
83 2a	jsr -60(a6)	165 126	rtss	249 yK	movea.l _ExtIntuiBase,a6
84 2.	movea.l (a7)+,a6	166 127	rtss	250 V1	move.l 12(a7),d0
85 2H	rtss	167 128	movea.l _ExtIntuiBase,a6	251 2C	movea.l 8(a7),a0
86 xC0	_RemWindow.	168 129	movea.l 8(a7),a0	252 2K	jsr -204(a6)
87 9V3	move.l a6,-(a7)	169 130	jsr -138(a6)	253 2F	movea.l a7+,a6
88 M8	movea.l _ExtIntuiBase,a6	170 131	movea.l (a7)+,a6	254 2D	rtss
89 2d	movea.l 8(a7),a0	171 132	rtss	255 2p0	_CrangeProp
90 I	jsr -66(a6)	172 133	rtss	256 2F3	move.l a6,-a7
91 v2	movea.l a7+,a6	173 134	rtss	257 6e	movea.l _ExtIntuiBase,a6
92 X2	rtss	174 135	movea.l _ExtIntuiBase,a6	258 MN	movea.l 8(a7),a0
93 2a3	move.l a6,-(a7)	175 136	movea.l 8(a7),a0	259 Ym	movea.l 12(a7),d0-d4
94 1F	movea.l _ExtIntuiBase,a6	176 137	jsr -144(a6)	260 9Q	jsr -210(a6)
95 XJ	movea.l 8(a7),d0-d2	177 138	movea.l (a7)+,a6	261 gX	movea.l (a7)+,a6
96 K5	jsr -72(a6)	178 139	rtss	262 1J	rtss
97 2t	movea.l a7+,a6	179 140	MenuSubIter:	263 DT0	_ShowRequester.
98 2f	rtss	180 A40	movea.l a6,-a7	264 1N3	move.l a6,-(a7)
99 2p0	_PutEllipse:	181 223	movea.l _ExtIntuiBase,a6	265 2D	movea.l _ExtIntuiBase,a6
100 Nj2	move.l a6,-(a7)	182 21	movea.l _ExtIntuiBase,a6	266 2M	movea.l 8(a7),d0-d4
101 2M	movea.l _ExtIntuiBase,a6	183 9A	movea.l 8(a7),a0	267 QJ	jsr -216(a6)
102 2D	movea.l 8(a7),d0-d3	184 2V	jsr -150(a6)	268 2e	movea.l (a7)+,a6
103 v2	jsr -78(a6)	185 5J	movea.l (a7)+,a6	269 PQ	rtss
104 90	movea.l (a7)+,a6	186 43	rtss	270 200	_RequesterEnd:
105 2n	rtss	187 2W0	rtss	271 2U3	move.l a6,-(a7)
106 2W0	_PutLine:	188 233	movea.l a6,-a7	272 17	movea.l _ExtIntuiBase,a6
107 2d3	move.l a6,-(a7)	189 2	movea.l _ExtIntuiBase,a6	273 2e	movea.l 8(a7),a0
108 2T	movea.l _ExtIntuiBase,a6	190 GR	movea.l 8(a7),a0	274 2e	jsr -222(a6)
109 2K	movea.l 8(a7),d0-d3	191 2	jsr -56(a6)	275 2L	movea.l (a7)+,a6
110 2a	jsr -84(a6)	192 2Q	movea.l (a7)+,a6	276 WX	rtss
111 27	movea.l (a7)+,a6	193 2	rtss	277 2V0	Message:
112 2t	rtss	194 20	SetMenuItem:	278 Fb3	move.l a6,-(a7)
113 210	_PutPoint	195 25	movea.l a6, a7	279 SE	movea.l _ExtIntuiBase,a6
114 2x3	move.l a6,-(a7)	196 21	movea.l _ExtIntuiBase,a6	280 1j	movea.l 8(a7),a0
115 2a	movea.l _ExtIntuiBase,a6	197 20	movea.l 8(a7),a0	281 17	jsr -228(a6)
116 2L	movea.l 8(a7),d0-d1	198 21	jsr -162(a6)	282 2a	movea.l (a7)+,a6
117 2D	jsr -90(a6)	199 gX	movea.l (a7)+,a6	283 2e	rtss
118 2E	movea.l (a7)+,a6	200 2	rtss	284 2G0	Class:
119 2D	rtss	201 20	GetPropW:	285 213	move.l a6,-a7
120 2y0	SetColor:	202 233	movea.l a6, a7	286 2L	movea.l _ExtIntuiBase,a6
121 2a3	move.l a6, (a7)	203 2	movea.l _ExtIntuiBase,a6	287 2B	jsr -234(a6)
122 2h	movea.l _ExtIntuiBase,a6	204 2	movea.l 8(a7),d0-d4	288 2y	movea.l (a7)+,a6
123 2v	movea.l 8(a7),d0-d2	205 2	jsr -168(a6)	289 2K	rtss
124 25	jsr -96(a6)	206 2e	movea.l (a7)+,a6	290 2Q0	Code:
125 2L	movea.l a7+,a6	207 PQ	rtss	291 2a3	move.l a6,-(a7)
126 27	rtss	208 210	SetGadget:	292 2e	movea.l _ExtIntuiBase,a6
127 2N0	Mode	209 23	movea.l a6,-a7	293 2V	jsr -240(a6)
128 233	move.l a6,-(a7)	210 21	movea.l _ExtIntuiBase,a6	294 2	movea.l (a7)+,a6
129 2e	movea.l _ExtIntuiBase,a6	211 2Y	movea.l 8(a7),d0-d5/a0-a2	295 2V	rtss
130 2S	movea.l 8(a7),d0	212 2N	jsr -174(a6)	296 2W0	_Cadge'id
131 2E	jsr 102(a6)	213 2	movea.l (a7)+,a6	297 2y3	move.l a6,-(a7)
132 25	movea.l (a7)+,a6	214 2K	rtss	298 2X	movea.l _ExtIntuiBase,a6
133 2E	rtss	215 20	rtss	299 2K	jsr -246(a6)
134 2B0	GetPosX.	216 233	movea.l a6, a7	300 2A	movea.l a7+,a6
135 213	move.l a6,-(a7)	217 2E	movea.l _ExtIntuiBase,a6	301 2V	rtss
		218 22	movea.l 12(a7),a0	302 2N0	MenuId
		219 2K	movea.l 8(a7),a0	303 203	move.l a6,-(a7)


```

304 r4 movea.l _extintuiBase,a6
305 Ob jsr -252(a6)
306 P0 movea.l (a7)+,a6
307 12 rts
308 ko0 Itemid:
309 k63 move.l a6,-(a7)
310 x3 movea.l _ExtintuiBase,a6
311 4t jsr -258(a6)
312 V4 movea.l (a7)+,a6

```

```

313 76 rts
314 ty0 SubId
315 qU3 move.l a6,-(a7)
316 3p movea.l ExtintuiBase,a6
317 rr jsr -264(a6)
318 bS movea.l (a7)+,a6
319 DE rts
320 st0 _SetPal.
321 W13 move.l a6,-(a7)

```

```

322 9w movea.l ExtintuiBase,a6
323 2t movea.l a7,-(a6)
324 f2 jsr -270(a6)
325 1Z movea.l (a7)+,a6
326 KL rts
327 f8 end
(C) 1988 M&T

```

Listing 2. (Schluß)

Programmname	extintui.lib.fd
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	ASCII-Text
Bemerkung:	Siehe Text
Programmautor: Jürgen Haage <hr/> <pre> 1 rm0 * extintui.lib.fd, v1.0 2 VO * Jürgen Haage, Burgstraße 30 3 Xa * 6552 Bad Münster am Stein 4 br ##base _extintuibase 5 %0 ##bias 30 6 32 ##public 7 /x FinLib 8 Ob ShowWindow(left,top,width,height,idemp,flags,titel)(D0,D1,D2 ,D3,D4,D5,A0) 9 eU ShowScreen(depth,mode,titel)(D0,D1,A0) 10 PK ActScreen(screen)(A0) 11 o9 RemScreen(screen)(A0) 12 sh ActWindow(window)(A0) 13 36 RemWindow(window)(A0) 14 WL PutCircle(x,y,radius)(D0,D1,D2) 15 Ux PutEllipse(x,y,vradius,hradius)(D0,D1,D2,D3) 16 ma PutLine(x1,y1,x2,y2)(D0,D1,D2,D3) 17 eM PutPoint(x,y)(D0,D1) 18 Fr SetColor(front,back,out)(D0,D1,D2) 19 Gp Mode(mode)(D0) 20 YZ GetCsr 21 oY GetPosY()() 22 am PrintText(text,x,y)(A0,D0,D1) </pre>	

```

23 rm MouseX()()
24 qq MouseY()()
25 BE MenuHeader(text)(A0)
26 gH MenuItem(text)(A0)
27 oE MenuSubItem(text)(A0)
28 av ...
29 Aa SetStringInfo(buffer)(A0)
30 lB SetPropInfo(flags,hpot,vpot,hbody,vbody)(D0,D1,D2,D3,D4)
31 mZ SetGadget(left,top,width,height,active,type,text,info,id)(D0
,D1,D2,D3,D4,D5,A0,A1,A2)
32 lO GetStrInfo(req,id)(A0,D0)
33 8h GetPropX(req,id)(A0,D0)
34 Dn GetPropY(req,id)(A0,D0)
35 6e GetPropW(req,id)(A0,D0)
36 9S GetPropH(req,id)(A0,D0)
37 KS ChangeProp(req,id,x,y,w,r)(A0,D0,D1,D2,D3,D4)
38 rR ShowRequester(left,top,width,height,color)(D0,D1,D2,D3,D4)
39 Pa RequesterEnd(req)(A0)
40 PK Message(window)(A0)
41 Kb Class()()
42 cZ Code()()
43 Je GadgetId()()
44 8o MenuId()()
45 w6 ...
46 A1 ...
47 7F SetPal(req,red,green,blue)(D0,D1,D2,D3)
48 lQ ##end
(C) 1988 M&T

```

Listing 3. Basic-Programmierer generieren aus »extintui.lib.fd« eine »bmap«-Datei

Programmname:	Vorspann
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	Amiga-Basic 1.2
Bemerkung:	Siehe Text
Programmautor: Jürgen Haage <hr/> <pre> 1 H\$4 DECLARE FUNCTION ShowScreen\$(Tiefe,Modus,Titel) LIBRARY 2 X\$0 REM DECLARE FUNCTION ActScreen(screen) LIBRARY 3 W\$ REM DECLARE FUNCTION RemScreen(screen) LIBRARY 4 hL4 DECLARE FUNCTION ShowWindow\$(LeftEdge,TopEdge,WIDTH,Height,Idemp,Flags,Titel) LIBRARY 5 Vr0 REM DECLARE FUNCTION ActWindow(vWindow) LIBRARY 6 u\$ REM DECLARE FUNCTION RemWindow(window) LIBRARY 7 Cr REM DECLARE FUNCTION PutCircle(x,y,radius) LIBRARY 8 w0 REM DECLARE FUNCTION PutEllipse(x,y,vradius,hradius) LIBRARY 9 dM REM DECLARE FUNCTION PutLine (x1,y1,x2,y2) LIBRARY 10 3\$ REM DECLARE FUNCTION PutPoint(x,y) LIBRARY 11 Uf REM DECLARE FUNCTION PrintText(Text,x,y) LIBRARY 12 wA REM DECLARE FUNCTION SetColor(front,back,out) LIBRARY 13 Bl REM DECLARE FUNCTION Mode(DrawMode) LIBRARY 14 z44 DECLARE FUNCTION GetPosX() LIBRARY 15 5B DECLARE FUNCTION GetPosY() LIBRARY 16 k0 DECLARE FUNCTION MouseX() LIBRARY 17 p6 DECLARE FUNCTION MouseY() LIBRARY 18 4z DECLARE FUNCTION MenuHeader\$(Textadresse) LIBRARY 19 UN DECLARE FUNCTION MenuItem\$(Textadresse) LIBRARY 20 Fi DECLARE FUNCTION MenuSubItem\$(Textadresse) LIBRARY 21 Hx0 REM DECLARE FUNCTION KillMenu(window) LIBRARY 22 LW\$ DECLARE FUNCTION SetStringInfo\$(maxchars) LIBRARY 23 e4 DECLARE FUNCTION SetPropInfo\$(F,aga,hpot,vpot,hbody,vbody) LIBRARY </pre>	

```

24 37 DECLARE FUNCTION SetGadget$(LeftEdge,TopEdge,WIDTH,Height,Activ,Typ,Text,Info,Id) LIBRARY
25 NJ DECLARE FUNCTION GetStrInfo$(requester,Id) LIBRARY
26 P0 DECLARE FUNCTION GetPropX$(requester,Id) LIBRARY
27 W3 DECLARE FUNCTION GetPropY$(requester,Id) LIBRARY
28 LO DECLARE FUNCTION GetPropW$(requester,Id) LIBRARY
29 uX DECLARE FUNCTION GetPropH$(requester,Id) LIBRARY
30 2B0 REM DECLARE FUNCTION ChangeProp(requester,Id,hpot,vpot,hbody,vbody) LIBRARY
31 Jv4 DECLARE FUNCTION ShowRequester$(LeftEdge,TopEdge,WIDTH,Height,Farbe) LIBRARY
32 FS DECLARE FUNCTION RequesterEnd$(requester) LIBRARY
33 Dk DECLARE FUNCTION Message$(window) LIBRARY
34 hz DECLARE FUNCTION Class$() LIBRARY
35 6a DECLARE FUNCTION Code$() LIBRARY
36 mF DECLARE FUNCTION GadgetId$() LIBRARY
37 dT DECLARE FUNCTION MenuId$() LIBRARY
38 Tw DECLARE FUNCTION ItemId$() LIBRARY
39 eo DECLARE FUNCTION SubId$() LIBRARY
40 Gm0 REM DECLARE FUNCTION SetPal(RegNr,red,green,blue)
41 gl LIBRARY "extintui.library"
42 6C ON ERROR GOTO Close
43 QT ON BREAK GOSUB CloseA1
44 gt BREAK ON
45 eX REM ***** Hier das Hauptprogramm *****
46 LG1 GOTO CloseAll
47 VQ0 END
48 n9 CloseAll.
49 WJ2 IF ERR=150 THEN CLS.PRINT "Fehler bei Bibliotheksaufruf".F
OR r=0 TO 2000 NEXT r
50 Sg LIBRARY CLOSE
51 L2 BREAK OFF
52 ux ON ERROR GOTO 0
53 bW0 END
(C) 1988 M&T

```

Listing 4. In dieses Listing bauen Sie ein Basic-Programm ein

Programmname:	extintui_fcn.c
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	cc extintui_fcn.c +L +B -S ln -s extintui.library extintui.lib.o extintui_fcn.o -LCL32
Bemerkung:	Funktionen der extintui.library

Programmautor Jürgen Haage

```

1 Uy0 /*****/
2 m8 /* extintui_fcn.c v1.0 */
3 4s /*
4 I7 /* Jürgen Haage Burgstraße 30
5 8b /* 6952 Bad Münster a. St. Ebernburg 2
6 Z3 /*****/
7 d6 #include "intuition/intuition.h"
8 Kv #include "graphics/gfxmacros.h"
9 Un #include "exec/memory.h"
10 nn struct extintlib
11 7a2
12 FV struct Screen *actscreen,
13 4x struct Window *actwindow;
14 4N struct RastPort *actrp;
15 39 struct Gadget *gadget;
16 go struct IntuiMessage message,
17 aW },
18 aRO FindLib()
19 F1
20 52 struct Task *current;
21 BE while(!current = (struct Task *)FindTask(0)),
22 Ia return(current->to_UserData),
23 Na
24 ek ShowWindow(x,y,width,height,idcmp,flags,str)
25 8U SHORT x,y,width,height,
26 nC ULONG idcmp,flags;
27 ex BYTE *str,
28 On {
29 PE struct NewWindow newwindow;
30 li struct Window *window;
31 31 struct Screen *screen,
32 5y long *test,
33 da struct extintlib *user;
34 84 if(!user = (struct extintlib *)FindLib())
35 242 return(OL);
36 liQ if(user->actscreen)
37 X02 {
38 eK if((width + x) > user->actscreen->Width)
39 NG4 width = user->actscreen->Width - x;
40 6v2 if((height + y) > user->actscreen->Height)
41 Rg4 height = user->actscreen->Height - y;
42 gB2
43 WJ0 else
44 e72
45 wH if((width + x) > 640)
46 4D4 width = 640 - x;
47 Za2 if((height + y) > 256)
48 8W4 height = 256 - y;
49 n12
50 9R0 newwindow.LeftEdge = x,
51 lp newwindow.TopEdge = y;
52 CL newwindow.Width = width;
53 PE newwindow.Height = height;
54 a6 newwindow.DetailPen = 0,
55 XP newwindow.BlockPen = 1,
56 WC newwindow.CheckMark = 0L,
57 Je newwindow.MinWidth = 1C,
58 ps newwindow.MinHeight = 10,
59 WQ newwindow.MaxWidth = 640,
60 ju newwindow.MaxHeight = 256,
61 j1 newwindow.FirstGadget 0L,
62 VO newwindow.IDCMPFlags = idcmp,
63 A6 newwindow.Flags = flags,
64 TM newwindow.Title = str;
65 14 newwindow.Type = WBENCHSCREEN,
66 Om if(user->actscreen)
67 lb2

```

```

68 Je newwindow.Screen = (struct Screen *)user->actscreen,
69 W8 newwindow.Type = CUSTOMSCREEN,
70 8d
71 liQ if(window = (struct Window *)OpenWindow(&newwindow))
72 6Z2
73 Gw user->actwindow = (struct Window *)window;
74 gV user->actrp = (struct RastPort *)window->RPort,
75 XJ return(window),
76 E4
77 fK0 return(OL);
78 G4
79 NW ShowScreen(depth,mod,tit)
80 vH SHORT depth,
81 AQ USHORT mod,
82 QD BYTE *tit;
83 Hk
84 SP struct extintlib *user;
85 8P struct NewScreen *newscreen,
86 wU struct Screen *screen,
87 oK if((depth > 4 && (mod == 1 || mod == 3)) || (depth > 5 &&
mod == 2) || (depth > 6) || (mod > 3))
88 qu2 return(OL);
89 vJ0 if(!newscreen = (struct NewScreen *)AllocMem(sizeof(struct
NewScreen),MEMF_CLEAR)),
90 8x2 return(OL);
91 C50 newscreen->LeftEdge = 0;
92 8W newscreen->TopEdge = 0;
93 ao newscreen->Depth = depth,
94 8b newscreen->Width = 320;
95 8Q newscreen->Height = 256,
96 xI newscreen->DetailPen = 0;
97 GL newscreen->BlockPen = 1,
98 3g newscreen->ViewModes = 0;
99 hU if(mod > 0 && mod != 2)
100 YL2
101 hu newscreen->Width = 640,
102 H3 newscreen->ViewModes = HIRIS,
103 8A
104 r40 if(mod > 1)
105 862 {
106 1W newscreen->Height = 512,
107 ng newscreen->ViewModes |= LACE,
108 kF
109 rT0 if(mod==0 && depth==6
110 nT2 newscreen->ViewModes |= EXTRA_HALFBRIT;
111 VEO newscreen->Type = CUSTOMSCREEN; /* Customscreen */
112 RM newscreen->DefaultTitle = tit,
113 jL if(!user = (struct extintlib *)FindLib())
114 CL2 return(OL),
115 aW0 if(screen = (struct Screen *)OpenScreen(newscreen))
116 oH2 {
117 mC FreeMem(newscreen,sizeof(struct NewScreen));
118 j1 user->actscreen = (struct Screen *)screen;
119 ev return(screen);
120 wR
121 qQ0 FreeMem(newscreen,sizeof(struct NewScreen)),
122 GT return(OL);
123 2L {
124 Pn ActScreen(screen)
125 ZX struct Screen *screen,
126 yK
127 96 struct extintlib *user;
128 Q8 if(!screen)
129 6F2 return;
130 8eQ if(!user = (struct extintlib *)FindLib())
131 8h2 return;
132 8aQ ScreenToFront(screen),
133 yG user->actscreen = (struct Screen *)screen;
134 Af
135 bM RemScreen(screen)
136 li struct Screen *screen;
137 9a
138 Li struct Window *remwindow,*copywindow;
139 Li struct extintlib *user;
140 Li struct Gadget *gadget,*gadget2,
141 dE if(!screen)
142 Ja2 return;
143 2X0 if(remwindow = (struct Window *)screen->FirstWindow)
144 8J2
145 Mj do
146 J14
147 XP copywindow = (struct Window *)remwindow->NextWindow,
148 7D CloseWindow(remwindow);
149 tx remwindow = copywindow;

```



```

150 lw      ] while(remwindow);
151 rw2
152 gv0 CloseScreen(screen);
153 Na if(!user = (struct extintlib *)FindLib()))
154 V42      return;
155 Lb0 user->actscreen = 0L;
156 MX user->actwindow = 0L;
157 UT user->actrp = 0L;
158 Y3 {
159 xH ActWindow(window)
160 x7 struct Window *window;
161 x0
162 lf struct extintlib *user;
163 xA if(!window
164 xE2      return;
165 xB0 if(!user = (struct extintlib *)FindLib
166 hG2      return;
167 vY0 ActivateWindow(window);
168 xT user->actwindow = (struct Window *)window;
169 lF user->actrp = (struct RastPort *)window->RPort;
170 xF
171 or RemWindow(window)
172 xJ struct Window *window;
173 J0
174 ur struct extintlib *user;
175 xD struct Gadget *gadget,*gadget2;
176 CN if(!window)
177 xE2      return;
178 mD if(!user = (struct extintlib *)FindLib())
179 xI2      return;
180 lI0 if(window == user->actwindow
181 xK2
182 mx      user->actwindow = 0L;
183 xI      user->actrp = 0L;
184 xT
185 wT0 CloseWindow(window);
186 OV
187 xT PutCircle(x,y,radius)
188 MF SHORT x,y,radius,
189 xS
190 A7 struct extintlib *user;
191 xb if(!user = (struct extintlib *)FindLib())
192 xG2      return;
193 qh0 if(user->actrp)
194 fC2 DrawEllipse(user->actrp,x,y,radius,radius); /* window ra
      stport */
195 9e0
196 lI PutEllipse(x,y,vradius,hradius)
197 qu SHORT x,y,vradius,hradius;
198 xS {
199 JQ struct extintlib *user;
200 xk if(!user = (struct extintlib *)FindLib())
201 Gp2      return;
202 xq0 if(user->actrp
203 JNG DrawEllipse(user->actrp,x,y,vradius,hradius); /* window
      rastport */
204 lno {
205 xm PutLine(x1,y1,x2,y2)
206 qE SHORT x1,y1,x2,y2;
207 xk
208 SP struct extintlib *user;
209 ht if(!user = (struct extintlib *)FindLib())
210 ky2      return;
211 xEQ if(user->actrp)
212 Mp2 {
213 xJ Move(user->actrp,x1,y1);
214 xI Draw(user->actrp,x2,y2);
215 Ty }
216 Ue0 {
217 xH PutPoint(x,y)
218 lQ SHORT x,y;
219 TW
220 eb struct extintlib *user;
221 T5 if(!user = (struct extintlib *)FindLib())
222 hA2      return;
223 xB0 if(user->actrp
224 fI2
225 xw Move(user->actrp,x,y ,
226 OI Draw(user->actrp,x,y);
227 lA
228 gB0
229 b0 SetColor,front,back,out;
230 Ad BYTE front,back,out;
231 fS

```

```

232 en struct extintlib *user;
233 fh if(!user = (struct extintlib *)FindLib())
234 xM2      return;
235 WNO if(user->actrp)
236 xD2 {
237 xk SetAPen(user->actrp,front);
238 lI SetBPen(user->actrp,back);
239 OT SetOPen(user->actrp,out);
240 xH
241 x00
242 xS Mode mode
243 xS BYTE mode;
244 xI
245 x0 struct extintlib *user;
246 xJ if(!user = (struct extintlib *)FindLib())
247 xJ2      return;
248 xA0 if(user->actrp)
249 hO2 SetDrMd(user->actrp,mode);
250 xX0
251 DV GetPosX()
252 OT {
253 B8 struct extintlib *user;
254 Oe if(!user = (struct extintlib *)FindLib())
255 hB2      return;
256 xI0 if user->actrp
257 lV2      return(user->actrp->cp_x);
258 xF0 {
259 Oe GetPosY()
260 xB {
261 JQ struct extintlib *user;
262 xk if(!user = (struct extintlib *)FindLib())
263 Gp2      return;
264 xq0 if user->actrp
265 y92      return(user->actrp->cp_y);
266 lno {
267 xJ PrintText(text,x,y)
268 xQ BYTE *text,
269 qp SHORT x,y,
270 lI {
271 TQ struct extintlib *user;
272 lA if(!user = (struct extintlib *)FindLib())
273 xJ2      return;
274 x00 if(user->actrp)
275 xq2 {
276 xI Move(user->actrp,x,y),
277 xA Text(user->actrp,text,strlen(text));
278 Us }
279 V00
280 Qr MouseX()
281 lV {
282 eb struct extintlib *user;
283 T5 if(!user = (struct extintlib *)FindLib())
284 hA2      return;
285 xP0 if(user->actwindow)
286 y32      return(user->actwindow->MouseX);
287 xB0
288 a2 MouseY()
289 x4
290 xJ struct extintlib *user;
291 xD if(!user = (struct extintlib *)FindLib())
292 xJ2      return;
293 xX0 if(user->actwindow)
294 xJ2      return(user->actwindow->MouseY);
295 lG0
296 sa MenuHeader(text)
297 vt BYTE *text;
298 xB {
299 xk SHORT maxv = 0, l = 0;
300 wt struct extintlib *user;
301 XR struct Menu *menu,*newmenu,*oopymenu;
302 m0 if(!user = (struct extintlib *)FindLib())
303 xB2      return(0);
304 lq0 if(!user->actwindow)
305 xJ2      return(0);
306 qn0 if(!newmenu = (struct Menu *)AllocMem(sizeof(struct Menu),
      MEMF_CLEAR,
307 QI2      return(0);
308 xP0 Forbid();
309 x0 maxv = (strlen(text)*8)+8;
310 lZ if(menu = (struct Menu *)user->actwindow->MenuStrip)
311 xB2 {

```

Listing 5. Die Funktionen für die »extintui.library«.
Bitte mit dem Checksummer (Seite 143) eingeben.

```

312 B4 do
313 254
314 03 copymenu = (struct Menu *)menu;
315 B8 l = menu->LeftEdge;
316 92 l += menu->Width;
317 8b menu = (struct Menu *)menu->NextMenu;
318 11 | while(menu);
319 0y2 copymenu->NextMenu = (struct Menu *)newmenu;
320 A1
321 040 e.s+
322 xE2 user->actwindow->MenuStrip = (struct Menu *)newmenu;
323 A40 newmenu->LeftEdge = l;
324 02 newmenu->Width = maxv;
325 T3 newmenu->Height = 10;
326 Q8 newmenu->Flags = 1;
327 b2 newmenu->MenuName = text;
328 1 newmenu->FirstItem = 0L;
329 8y newmenu->NextMenu = 0L;
330 K5 SetMenuStrip(user->actwindow, user->actwindow->MenuStrip);
331 03 Permit();
332 F return 1;
333 0a
334 1J MenuItem(text);
335 X1 BYTE *text;
336 Kc
337 KJ struct extintlib *user;
338 Co struct Menu *menu;
339 av struct MenuItem *menuitem, *newmenuitem, *copymenuitem;
340 18 struct IntuiText *itext;
341 1p SHORT top = 0, width;
342 Q2 if(!user = (struct extintlib *)FindLib())
343 Y72 ret = -1;
344 PU0 if(!user->actwindow)
345 2x2 return(0);
346 s00 if(menu = (struct MenuItem *)user->actwindow->MenuStrip)
347 X02 {
348 PC while(menu->NextMenu)
349 074 menu = (struct Menu *)menu->NextMenu;
350 092
351 140 else
352 942 return(0);
353 VJ0 if(!newmenuitem = (struct MenuItem *)AllocMem(sizeof(struct MenuItem), MEMF_CLEAR))
354 B62 return(0);
355 Ld0 if(!itext = (struct IntuiText *)AllocMem(sizeof(struct IntuiText), MEMF_CLEAR))
356 g92
357 03 FreeMem(newmenuitem, sizeof(struct MenuItem));
358 FA return(0);
359 n1 }
360 390 itext->IText = (BYTE *)text;
361 Jq itext->DrawMode = 2;
362 m1 width = IntuiTextLength(itext);
363 v1 Forbid();
364 s8 if(menuitem = (struct MenuItem *)menu->FirstItem)
365 p12 {
366 v1 do
367 x84 {
368 nf copymenuitem = (struct MenuItem *)menuitem;
369 e1 if(width < menuitem->Width)
370 916 width = menuitem->Width;
371 0b4 else
372 k96 menuitem->Width = width;
373 1d4 top = menuitem->TopEdge;
374 2Q menuitem = (struct MenuItem *)menuitem->NextItem;
375 BK } while(menuitem);
376 122 top += 10;
377 w8 copymenuitem->NextItem = (struct MenuItem *)newmenuitem;
378 6b }
379 wJ0 else
380 6Q2 menu->FirstItem = (struct MenuItem *)newmenuitem;
381 900 newmenuitem->LeftEdge = 0;
382 40 newmenuitem->TopEdge = top;
383 15 newmenuitem->Width = width;
384 29 newmenuitem->Height = 10;
385 1c newmenuitem->Flags = (ITEMTEXT | ITEMENABLED | HIGHCOMP);
386 1m newmenuitem->ItemFill = (struct IntuiText *)itext;
387 37 newmenuitem->NextItem = 0L;
388 L4 newmenuitem->SelectFill = 0L;
389 0d newmenuitem->SubItem = 0L;
390 gA newmenuitem->NextSelect = 0xFFFF;
391 J5 SetMenuStrip(user->actwindow, user->actwindow->MenuStrip);

```

```

392 DR Permit();
393 00 return(1);
394 Xr ;
395 pC MenuSubItem(text)
396 WU BYTE *text;
397 1c
398 ZM struct extintlib *user;
399 Bn struct Menu *menu;
400 k1 struct MenuItem *menuitem, *copymenuitem, *submenuitem, *newsubitem;
401 m7 struct IntuiText *itext;
402 1f SHORT top = 0, width;
403 P1 if(!user = (struct extintlib *)FindLib())
404 X62 return;
405 OT0 if(!user->actwindow)
406 192 return(0);
407 WAO if(menu = (struct Menu *)user->actwindow->MenuStrip)
408 W22 {
409 Xr while(menu->NextMenu)
410 Wk4 menu = (struct Menu *)menu->NextMenu;
411 182
412 TGO else
413 832 return(0);
414 MY0 if(menuitem = (struct MenuItem *)menu->FirstItem)
415 062
416 u8 while(menuitem->NextItem)
417 s14 menuitem = (struct MenuItem *)menuitem->NextItem;
418 kF2 }
419 aN0 else
420 FA2 return(0);
421 Ag0 if(!newsubitem = (struct MenuItem *)AllocMem(sizeof(struct MenuItem), MEMF_CLEAR))
422 KC2 return(0);
423 250 if(!itext = (struct IntuiText *)AllocMem(sizeof(struct IntuiText), MEMF_CLEAR))
424 nF2 {
425 00 FreeMem(newsubitem, sizeof(struct MenuItem));
426 1G return(0);
427 t0 }
428 UAO itext->IText = (BYTE *)text;
429 pw itext->DrawMode = 2;
430 Gc width = IntuiTextLength(itext);
431 1D Forbid();
432 Gx if(submenuitem = (struct MenuItem *)menuitem->SubItem)
433 v02 {
434 10 do
435 xQ4 {
436 kU copymenuitem = (struct MenuItem *)submenuitem;
437 ZM if(width < submenuitem->Width)
438 bN6 width = submenuitem->Width;
439 uh4 else
440 FE6 submenuitem->Width = width;
441 g54 top = submenuitem->TopEdge;
442 3m submenuitem = (struct MenuItem *)submenuitem->NextItem;
443 19 } while(submenuitem);
444 rK2 top += 10;
445 JY copymenuitem->NextItem = (struct MenuItem *)newsubitem;
446 0r
447 2p0 else
448 nR2 menuitem->SubItem = (struct MenuItem *)newsubitem;
449 0M0 newsubitem->LeftEdge = menuitem->Width + 5;
450 Qq newsubitem->TopEdge = top;
451 Ps newsubitem->Width = width;
452 FA newsubitem->Height = 10;
453 JX newsubitem->Flags = (ITEMTEXT | ITEMENABLED | HIGHCOMP);
454 SE newsubitem->ItemFill = (struct IntuiText *)itext;
455 hm newsubitem->NextItem = 0L;
456 nf newsubitem->SelectFill = 0L;
457 RV newsubitem->SubItem = 0L;
458 1D newsubitem->NextSelect = 0xFFFF;
459 pX SetMenuStrip(user->actwindow, user->actwindow->MenuStrip);
460 JX Permit();
461 yu return(1);
462 Sx ;
463 eK KillMenu(window)
464 11 struct Window *window;
465 Ru {
466 04 struct Menu *menu, *copymenu;
467 k9 struct MenuItem *menuitem, *copymenuitem, *subitem, *copysubitem;
468 u5 if(!window)
469 a92 return;

```



```

470 ei0 Forbid();
471 3g if( menu = ( struct Menu *)window->MenuStrip)
472 Y12
473 e1 do
474 a34
475 LZ if( menuitem = ( struct MenuItem *)menu->FirstItem)
476 ^56
477 15 do
478 e73
479 g88 if( subitem = ( struct MenuItem *)menuitem->SubItem)
480 g95
481 m9 do
482 187
483 74C copysubitem = ( struct MenuItem *)subitem->NextItem;
484 3j FreeMem( subitem,sizeof( struct IntuiText ));
485 w7 FreeMem( subitem,sizeof( struct MenuItem ));
486 ^h subitem = ( struct MenuItem *)copysubitem;
487 187 while( subitem),
488 ^h5 {
489 JR8 copymenuitem = ( struct MenuItem *)menuitem->NextItem;
490 3j FreeMem( menuitem,sizeof( struct IntuiText ));
491 Mf FreeMem( menuitem,sizeof( struct MenuItem ));
492 LX menuitem = ( struct MenuItem *)copymenuitem;
493 193 } while( menuitem);
494 1T6
495 114 copymenu = ( struct Menu *)menu->NextMenu;
496 .1 FreeMem( menu,sizeof( struct Menu ));
497 Ra menu = ( struct Menu *)copymenu;
498 66 } while( menu);
499 1X2 }
500 pt0 window->MenuStrip = 0L;
501 1C Permit ;
502 6b
503 8D SetStringInfo(buffer)
504 3R SHORT buffer;
505 5Y
506 34 struct StringInfo *stringinfo;
507 1A if(stringinfo = (struct StringInfo *)AllocMem(sizeof(struct StringInfo),MEMF_CLEAR))
508 8b2
509 78 if(stringinfo->Buffer = (BYTE *)AllocMem(((buffer+1)*2),MEMF_CLEAR))
510 4d4
511 3g stringinfo->UndoBuffer = (BYTE *) (stringinfo->Buffer)+buffer+1;
512 Ae stringinfo->MaxChars = buffer;
513 14 stringinfo->DispPos = 0;
514 4J stringinfo->BufferPos = 0;
515 1C return((struct StringInfo *)stringinfo);
516 1p
517 1a2 FreeMem(stringinfo,sizeof(struct StringInfo));
518 1R
519 ns0 return(0L);
520 0b {
521 3W SetPropInfo(flags,hbody,vbody,hpot,vpot)
522 bS JSHORT flags,hbody,vbody,hpot,vpot;
523 Nq
524 1W struct PropInfo *propinfo;
525 13 if(propinfo = (struct PropInfo *)AllocMem(sizeof(struct PropInfo),MEMF_CLEAR))
526 3.2
527 8A flags |= AUTOKNOB;
528 12 propinfo->Flags = flags;
529 RA propinfo->HorizPot = hpot;
530 16 propinfo->VertPot = vpot;
531 xt propinfo->HorizBody = hbody;
532 Ps propinfo->VertBody = vbody;
533 1b return((struct PropInfo *)propinfo);
534 07
535 380 return(0L);
536 e9 }
537 e1 SetGadget(x,y,extra,flag,typ,text,info,id)
538 Mg SHORT x,y,weite,höhe;
539 Nc USHORT typ,flag,id,
540 16 LONG info,
541 1p BYTE *text;
542 g9 {
543 0d struct Gadget *newgadget,*gadget,*copygadget;
544 52 struct IntuiText *itext,

```

```

545 tq struct extintlib *user;
546 XR struct Border *border;
547 3s SHORT err = 0,*array;
548 AT USHORT fl;
549 1N if(!user = (struct extintlib *)FindLib())
550 tS2 return;
551 Ls0 if(user->actwindow)
552 KP2 return(0L);
553 710 if(newgadget = (struct Gadget *)AllocMem(sizeof(struct Gadget),MEMF_CLEAR))
554 sL2
555 Ku if(border = (struct Border *)AllocMem(sizeof(struct Border),MEMF_CLEAR))
556 14 {
557 a2 if(array = (SHORT *)AllocMem(24,MEMF_CLEAR))
558 wP6 {
559 Hq if(itext = (struct IntuiText *)AllocMem(sizeof(struct IntuiText),MEMF_CLEAR))
560 yR8 {
561 Vq newgadget->GadgetRender = (struct Border *)border;
562 0a newgadget->GadgetText = (struct IntuiText *)itext;
563 LY3 border->XY = (SHORT *)array;
564 6b8 }
565 1j6 else
566 2a8 err = 1;
567 9a6 {
568 2a4 else
569 2d6 err = 1;
570 1b4
571 2p2 else
572 1g4 err = 1,
573 Pk2 }
574 010 if(err
575 Dg2
576 19 if(newgadget
577 014 FreeMem(newgadget,sizeof(struct Gadget));
578 1A2 if(array,
579 014 FreeMem(array,24);
580 S82 if(border)
581 2K4 FreeMem(border,sizeof(struct Border));
582 m2 return(0);
583 1
584 Ur0 Forbid();
585 pk if(gadget = (struct Gadget *)user->gadget)
586 2r2
587 1 do
588 114
589 nE copygadget = (struct Gadget *)gadget;
590 1v gadget = (struct Gadget *)gadget->NextGadget;
591 14 { while(gadget);
592 162 copygadget->NextGadget = (struct Gadget *)newgadget;
593 14
594 100 else
595 112 user->gadget = (struct Gadget *)newgadget;
596 820 *array = (SHORT)-1;
597 0g *(array+1) = (SHORT)-1;
598 1v *(array+2) = (SHORT)-1;
599 1w *(array+3) = (SHORT)höhe;
600 1b *(array+4) = (SHORT)weite;
601 dA *(array+5) = (SHORT)höhe;
602 1Y *(array+6) = (SHORT)weite;
603 1P *(array+7) = (SHORT)-1;
604 1w *(array+8) = (SHORT)-1;
605 1h *(array+9) = (SHORT)-1;
606 1h border->DrawMode = 2;
607 11 border->DrawMode = 2;
608 1c border->NextBorder = 0L;
609 1p typ |= REQGADGET;
610 11 itext->DrawMode = 2;
611 1v itext->IText = text;
612 06 if(typ == (REQGADGET|STRGADGET))
613 142 itext->LeftEdge = (IntuiTextLength(itext) + 5) * -1;
614 1W0 else
615 02 itext->LeftEdge = (weite - IntuiTextLength(itext)) / 2;
616 1x0 if(typ == (REQGADGET|PROPGADGET))
617 sm2 itext->TopEdge = -11;
618 na0 else
619 s12 itext->TopEdge = höhe - 5 / 2;
620 320 itext->NextText = 0L;
621 0g itext->ITextFont = 0L;
622 NG newgadget->LeftEdge = 1,

```

Listing 5. (Fortsetzung)

```

623 Qe newgadget->TopEdge = y;
624 u4 newgadget->Width = width;
625 d0 newgadget->Height = height;
626 HJ flag = 0;
627 J0 newgadget->Flags = GADDFLAGS;
628 q7 newgadget->Activation = flag;
629 Qb newgadget->GadgetType = type;
630 R1 newgadget->GadgetID = id;
631 n2 newgadget->NextGadget = 0;
632 Vq newgadget->SpecialInfo = 0;
633 xh if((type == (REQGADGET|STRGADGET)) && info)
634 2n2 newgadget->SpecialInfo = (struct StringInfo *)info;
635 4r0 else
636 4N2 if((type == (REQGADGET|PROPGADGET)) && info)
637 J4 newgadget->SpecialInfo = (struct PropInfo *)info;
638 7q2 else
639 q4 newgadget->GadgetType = (REQGADGET|BOULGADGET);
640 d80 Permit();
641 2V return(newgadget,
642 M);
643 1g GetStrInfo(requester, id,
644 V4, struct Requester *requester,
645 B2, SHORT id;
646 M);
647 7Z struct Gadget *gadget;
648 1F if(!requester)
649 ty2 return 0;
650 J60 else
651 g72 if(!gadget = (struct Gadget *)requester->ReqGadget))
652 w14 return(0);
653 QJ0 while(gadget->GadgetID != id && gadget)
654 ex2 gadget = (struct Gadget *)gadget->NextGadget;
655 VU0 if(gadget->GadgetType & STRGADGET)
656 XL2 return(gadget->SpecialInfo->Buffer);
657 QD0 else
658 272 return(0);
659 d80 }
660 7G GetPropX(requester, id)
661 mY struct Requester *requester;
662 Bq SHORT id;
663 d6 {
664 Qq struct Gadget *gadget;
665 1w if(!requester)
666 D82 return(0);
667 aN0 else
668 xD2 if(!gadget = (struct Gadget *)requester->ReqGadget))
669 QB4 return(0);
670 Fa0 while(gadget->GadgetID != id && gadget)
671 vE2 gadget = (struct Gadget *)gadget->NextGadget;
672 SJ0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
673 ex2 return(gadget->SpecialInfo->HorizPot);
674 hU0 else
675 M82 return(0);
676 aP0
677 ea GetPropY(requester, id)
678 Jp struct Requester *requester;
679 J7 SHORT id;
680 dN {
681 17 struct Gadget *gadget;
682 2D if(!requester)
683 UP2 return(0);
684 re0 else
685 EF2 if(!gadget = (struct Gadget *)requester->ReqGadget))
686 AS4 return(0);
687 wX0 while(gadget->GadgetID != id && gadget)
688 QV2 gadget = (struct Gadget *)gadget->NextGadget;
689 jK0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
690 b62 return(gadget->SpecialInfo->VertPot);
691 y10 else
692 dY2 return(0);
693 Bg0
694 nJ GetPropW(requester, id)
695 K6 struct Requester *requester;
696 QD SHORT id;
697 Bq {
698 wO struct Gadget *gadget;
699 qU if(!requester)
700 1g2 return(0);
701 Sv0 else
702 V42 if(!gadget = (struct Gadget *)requester->ReqGadget))
703 qJ4 return(0);
704 D80 while(gadget->GadgetID != id && gadget)
705 Tm2 gadget = (struct Gadget *)gadget->NextGadget;
706 Ob0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))

```

```

707 Rh2 return(gadget->SpecialInfo->HorizBody);
708 F20 else
709 up2 return(0);
710 Sx0
711 6n GetPropH(requester, id)
712 bN struct Requester *requester;
713 Hf SHORT id;
714 Sv {
715 Df struct Gadget *gadget;
716 71 if(!requester)
717 2x2 return(0);
718 P00 else
719 mD2 if(!gadget = (struct Gadget *)requester->ReqGadget))
720 S04 return(0);
721 UP0 while(gadget->GadgetID != id && gadget)
722 k32 gadget = (struct Gadget *)gadget->NextGadget;
723 H80 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
724 9t2 return(gadget->SpecialInfo->VertBody);
725 WJ0 else
726 B62 return(0);
727 J20 }
728 Ss ChangeProp(requester, id, x, y, w, h)
729 sa struct Requester *requester;
730 1w SHORT id;
731 4g USHORT x, y, w, h;
732 kD {
733 Vx struct Gadget *gadget;
734 x1 struct PropInfo *prop;
735 44 if(!requester)
736 tS2 return;
737 iV0 else
738 5W2 if(!gadget = (struct Gadget *)requester->ReqGadget))
739 wV4 return;
740 n10 while(gadget->GadgetID != id && gadget)
741 3M2 gadget = (struct Gadget *)gadget->NextGadget;
742 aB0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
743 vO2 {
744 2h prop = (struct PropInfo *)gadget->SpecialInfo;
745 vT2 prop->HorizPot = x;
746 9Q prop->VertPot = y;
747 rY prop->HorizBody = w;
748 V8 prop->VertBody = h;
749 d RefreshGList(gadget, requester->FWindow, requester, 1
750 b);
751 7c0
752 ET ShowRequester(x, y, w, h, farbe)
753 Tf SHORT x, y, w, h, farbe;
754 6Z {
755 yD struct extintlib *user;
756 J3 struct Requester *requester;
757 7J if(!user = (struct extintlib *)FindLib())
758 ej2 return(0);
759 hL0 if(!user->actwindow)
760 g12 return(0);
761 S00 if(!requester = (struct Requester *)AllocMem(sizeof(struct
Requester), MEMF_CLEAR))
762 in2 return(0);
763 4e0 requester->ReqGadget = user->gadget;
764 s7 requester->LeftEdge = x;
765 hq requester->TopEdge = y;
766 Ao requester->Width = w;
767 1o requester->Height = h;
768 1m requester->BackFill = farbe;
769 FQ if(Request(requester, user->actwindow))
770 v12 return(requester);
771 G90 else
772 ex2 return(0);
773 Ty0 }
774 yV RequesterEnd(requester)
775 aO struct Requester *requester;
776 Sv {
777 ev struct Gadget *gadget, *gadget2;
778 HB struct Border *border;
779 NQ struct StringInfo *strinfo;
780 g4 struct extintlib *user;
781 XF USHORT type;
782 W8 if(!user = (struct extintlib *)FindLib())
783 J82 return(0);
784 y20 if(requester)
785 J82 EndRequest(requester, requester->RWindow);
786 420 if(gadget = (struct Gadget *)user->gadget)
787 d62 do
788 J6 {

```



```

790 Kp gadget2 = (struct Gadget *)gadget;
791 2q gadget = (struct Gadget *)gadget2->NextGadget;
792 6w border = (struct Border *)gadget2->GadgetRender;
793 wv FreeMem(border->XY,24);
794 T1 FreeMem(border,sizeof(struct Border));
795 Ye FreeMem(gadget2->GadgetText,sizeof(struct IntuiText));
796 BW type = gadget2->GadgetType;
797 Rf if((type == (REQGADGET1 STRGADGET)) && gadget2->Special
Info)
{
798 oh6 struct StringInfo *)gadget2->SpecialInfo;
799 6e FreeMem(stringinfo->Buffer,((stringinfo->MaxChars) * 2));
800 4e FreeMem(stringinfo,sizeof(struct StringInfo));
801 xA }
802 wR }
803 a24 else
804 296 if((type == (REQGADGET1 PROPGADGET)) && gadget2->Spec
ialInfo)
FreeMem(gadget2->SpecialInfo,sizeof(struct PropInf
o));
805 eg8 FreeMem(gadget2,sizeof(struct Gadget));
806 v04 FreeMem(gadget2,sizeof(struct Gadget));
807 LW }
808 8X while(gadget);
809 JY2 }
810 7w0 user->gadget = 0L;
811 5e }
812 F6 Message(window)
813 0e struct Window *window;
814 4X }
815 F0 struct extintlib *user;
816 1H struct IntuiMessage *nachricht;
817 5h if(!user = (struct extintlib *)FindLib()))
818 fa2 return(0);
819 Zk0 if(!window)
820 892 if(!window = (struct Window *)user->actwindow))
821 1d4 return(0);
822 hh0 nachricht = (struct IntuiMessage *)GetMsg(window->UserPort
);
823 Qy if(nachricht)
824 2h2 {
825 k7v movmem(nachricht,&(user->message),sizeof(struct IntuiMes
sage));
826 JC ReplyMsg(nachricht);
827 6e return(1);
828 Mr }
829 q10 return(0);
830 Ot }
831 pv Cclass )
832 Mp {
833 Xv struct extintlib *user;
834 My if( user = (struct extintlib *)FindLib()))
835 w22 return(0);
836 ON0 return(user->message.Class);
837 V0 }
838 UA Code()
839 Tv {
840 ab struct extintlib *user;
841 T5 if(!user = (struct extintlib *)FindLib()))
842 3y2 return(0);
843 wy0 return(user->message.Code);
844 c7 }
845 y0 Gadgetid()
846 a3 {
847 11 struct extintlib *user;
848 Mo struct Gadget *gadget;
849 bD if(!user = (struct extintlib *)FindLib()))
850 B62 return(0);
851 Qf0 gadget = (struct Gadget *)user->message.IAddress;
852 nD return(gadget->GadgetID);
853 10 }
854 go Menuid()
855 j0 {
856 ur struct extintlib *user;
857 jL if(!user = (struct extintlib *)FindLib()))
858 JE2 return(0);
859 pv0 return((user->message.Code & 0x1F));
860 aN }
861 AC Itemid()
862 qJ {
863 1y struct extintlib *user;
864 qS if(!user = (struct extintlib *)FindLib()))
865 QL2 return(0);
866 190 return((user->message.Code >> 5) & 0x003F);
867 xD }
868 BV Subid()

```

```

869 xQ }
870 85 struct extintlib *user;
871 x2 if(!user = (struct extintlib *)FindLib()))
872 XS2 return(0);
873 rB0 return((user->message.Code >> 11) & 0x001F);
874 6b {
875 Yw SetPal(regnr,red,green,blue);
876 OX USHORT regnr,red,green,blue;
877 5Y {
878 1g struct Screen *screen;
879 HE struct extintlib *user;
880 6A if(!user = (struct extintlib *)FindLib()))
881 Em2 return;
882 YM0 if(red<0 || red>15 || green<0 || green>15 || blue<0 ||
blue>15 || regnr<0 || regnr>31)
883 Gp3 return;
884 m20 if(screen=(struct Screen *)user->actscreen)
885 4x3 SetRGB4 &screen->viewPort,regnr,red,green,blue ;
886 In0 }
(C) 1989 M&T

```

Listing 5. (Schluß)

Programmname:	M2Extintui.def
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	Modula-2
Compiler:	M2Amiga
Bemerkung:	siehe Text

Programmautor: Andreas Görtler

```

1 SY0 DEFINITION MODULE M2Extintui[ "extintui.library",1];
2 j4 (* by ag 11.12.88 3:00 *)
3 eP (* 1.3 = Includefiles.h für Version 1.3
4 oY3 M2 = M2-Amiga Modula-II
5 TF fen.o = Die Implementation der Routinen (extintui_fen.o)
6 X5 3 Kriterien: - TypenÜbereinstimmung mit 1.3(Include),M2,
und fen.o
7 bXQ - Rückgabewerte
8 1Z - Spezifikation der Parameter: z.B. WindowP
tr anstatt ADDRESS *)
9 DU0 FROM SYSTEM IMPORT ADDRESS,
10 FS FROM Exec IMPORT UByte,
11 EO FROM Graphics IMPORT DrawModeSet;
12 rb FROM Intuition IMPORT ScreenPtr,WindowPtr,RequesterPtr,IDCMP
FFlagSet,PropInfoPtr,
13 toM PropInfoFlagSet,ActivationFlagSet,Str
ingInfoPtr,GadgetPtr,
14 wh WindowFlagSet,
15 p20 (*----- Die Screen- und Window-Prozeduren -----
*)
16 qF PROCEDURE ShowScreen(depth[0]:INTEGER;
17 Z8L viewModes[1]:CARDINAL;
18 2P defaultTitle[8]:ADDRESS):ScreenPtr; GO
DE -42,
19 e22 (* if fail then return NIL else return ScreenPtr *)
20 RE (* erzeugt neuen Screen mit übergebenen Parametern *)
21 Mr0 PROCEDURE ActScreen(screenptr[8]:ScreenPtr); CODE -48;
22 MO PROCEDURE RemScreen(screenptr[8]:ScreenPtr); CODE -54;
23 Jy PROCEDURE ShowWindow(leftEdge[0]:INTEGER,
24 XPL topEdge[1]:INTEGER;
25 nr width[2]:INTEGER;
26 rv height[3]:INTEGER,
27 TQ idcmpFlags[4]:IDCMPFlagSet;
28 o1 flags[5]:WindowFlagSet;
29 KS title[8]:ADDRESS):WindowPtr; CODE -36;
30 pB2 (* if fail then return NIL else return WindowPtr *)
31 dQ0 PROCEDURE ActWindow(windowptr[8]:WindowPtr); CODE -60;
32 EJ PROCEDURE RemWindow(windowptr[8]:WindowPtr); CODE -66;
33 h2 (*----- Die Zeichenprozeduren -----
*)
34 IM PROCEDURE PutCircle(x[0]:INTEGER;
35 yX y[1]:INTEGER,
36 kw radius[2]:INTEGER); CODE -72,
37 Z10 PROCEDURE PutEllipse(x[0]:INTEGER,
38 1BE y[1]:INTEGER,

```

Listing 6. Das nötige Listing für Modula-2-Programmierer: »M2Extintui.def«

```

39 DQ      vradius[2]:INTEGER;
40 tk      hradius[3]:INTEGER; CODE -78;
41 9q2     (* Typen: s.o. ..radius bei allen SHORT *)
42 dy0 PROCEDURE PutLine(x1[0]:INTEGER,
43 eqI      y1[1]:INTEGER;
44 y        x2[2]:INTEGER;
45 FZ      y2[3]:INTEGER); CODE -84;
46 Gd0 PROCEDURE PutPoint(x[0]:INTEGER;
47 lqJ      y[1]:INTEGER); CODE -90;
48 vx2     (* setzt RastPort auf (x1,y1) bzw (x,y) und zeichnet mit
Graphics.Draw bis
49 275     (x2,y2, bzw (x,y).*)
50 OF0 PROCEDURE PrintText(textptr[8]:ADDRESS;
51 6GK      x 0 INTEGER,
52 4N      y[1]:INTEGER); CODE -120;
53 PF0 PROCEDURE Mode(drawMode[0]:DrawModeSet); CODE -102;
54 GT2     (* drawMode liefert Eintrag für aktuellen RastPort; Graph
ics.SetDrMd wird
55 Tv3     aufgerufen *)
56 1T2     (* drawMode in 1.3 und fon.c vom Typ BYTE, in M2 DrawMode
Set =
57 NO2     Bytegröße *)
58 d10 PROCEDURE SetColor(fgPen[0]:CARDINAL;
59 TB3     bgPen[1]:CARDINAL;
60 oy      aolPen[2]:CARDINAL); CODE -96;
61 re2     (* In M2 RastPort.fgPen, bgPen, aolPen : UByte; Die Argum
ente für Set_Pen :
62 2a2     CARDINAL; in 1.3 fgPen, bgPen, aolPen : BYTE; *)
63 bn0 PROCEDURE SetPal(n[0]:CARDINAL;
64 7CH      r[1]:CARDINAL;
65 qP      g[2]:CARDINAL;
66 6s      b[3]:CARDINAL); CODE -270;
67 qx2     (* entspricht Graphics.SetRGB4 *)
68 jt      (* Parameter sind in 1.3 und M2 USHORT (siehe ColorTable)
; fon.c : SHORT. *)
69 Ua0 PROCEDURE GetPosX():INTEGER; CODE -108;
70 Lq PROCEDURE GetPosY():INTEGER; CODE -114;
71 212     (* liefert RastPort.x bzw .y *)
72 P4      (* in M2 und 1.3 : SHORT *)
73 v30 PROCEDURE MouseX():INTEGER; CODE -126;
74 ou PROCEDURE MouseY():INTEGER; CODE -132;
75 Q32     (* liefert Window.mouseX bzw .mouseY *)
76 IX0     (* ----- Die Menüleisten -----
-----*)
77 Bq PROCEDURE MenuHeader(textptr[8]:ADDRESS):INTEGER; CODE -138
;
78 8V2     (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
Typ CARDINAL sein *)
79 W0      (* textptr liefert Eintrag für Menu.menuName; bei allen v
om Typ APTH *)
80 4T0 PROCEDURE MenuItem(textptr[8]:ADDRESS):INTEGER; CODE -144;
81 V12     (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
Typ CARDINAL sein *)
82 xt      (* textptr liefert Eintrag für IntuiText.iText; bei allen
vom Typ APTH *)
83 Kb0 PROCEDURE MenuSubItem(textptr[8]:ADDRESS):INTEGER; CODE -15
0;
84 1b2     (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
Typ CARDINAL sein *)
85 2v      (* textptr liefert Eintrag für IntuiText.iText; bei allen
vom Typ APTH *)
86 dy0 PROCEDURE KillMenu(windowptr[8]:WindowPtr); CODE -156;
87 2d      * ----- Gadgets und Requester -----
-----*)
88 pa PROCEDURE SetStringInfo(maxChars[8]:ADDRESS):StringInfoPtr;
CODE -162,
89 8v2     (* if fail then return NIL else return StringInfoPtr *)
90 gd0 PROCEDURE SetPropInfo(flags[0]:PropInfoFlagSet;
91 0OM      horizPot[1]:CARDINAL;
92 0d      vertPot[2]:CARDINAL;
93 0I      horizBody[3]:CARDINAL;
94 gv      vertBody[4]:CARDINAL); PropInfoPtr; CO
DE -168,
95 Wa2     (* if fail then return NIL else return new PropInfo with
APTH *)
(* In M2 und 1.3 und fon.c alle Parameter vom Typ USHORT
*)
97 170 PROCEDURE SetGadget(leftEdge[0]:INTEGER;
98 bK      topEdge[1]:INTEGER;
99 yJ      width[2]:INTEGER;
100 37      height[3]:INTEGER;
101 2Y      activation[4]:ActivationFlagSet;
102 17      gadgetType[5]:CARDINAL;
103 9a      gadgetText[8]:ADDRESS,

```

```

104 0I      specialInfo[9]:ADDRESS;
105 4K      gadgetID[10]:INTEGER):GadgetPtr; CODE -
174;
106 yL2     (* if actwindow = NIL then return NIL else Gadgetadresse
*)
107 vq      (* Alle Typen in fon.c und 1.3 o.k.; in M2 ist gadgetID v
om Typ SHORT *)
108 Q10 PROCEDURE GetStrInfo(requesterptr[8]:RequesterPtr;
109 wQ1      gadgetID[0]:INTEGER):ADDRESS; CODE -18
0;
110 Ag2     (* fail -> return NIL, sonst StringInfo.buffer *)
111 8M0 PROCEDURE GetPropH(requesterptr[8]:RequesterPtr;
112 4Q7      gadgetID[0]:INTEGER):CARDINAL; CODE -204
;
113 D42     (* fail -> return 0, sonst PropInfo.vertBody *)
114 D00 PROCEDURE GetPropW(req[8]:RequesterPtr;
115 AG1      gadgetID[0]:INTEGER):CARDINAL; CODE -198
;
116 jK2     (* fail -> return 0, sonst PropInfo.horizBody *)
117 KWO PROCEDURE GetPropX(req[8]:RequesterPtr;
118 wH2      gadgetID[0]:INTEGER):CARDINAL; CODE -186
;
119 pe2     (* fail -> return 0, sonst PropInfo.horizPot *)
120 Re0 PROCEDURE GetPropY(req[8]:RequesterPtr;
121 g13      gadgetID[0]:INTEGER):CARDINAL; CODE -192
;
122 A52     (* fail -> return 0, sonst PropInfo.vertPot *)
123 pE0 PROCEDURE ChangeProp(requesterptr[8]:RequesterPtr;
124 t1L      gadgetID[0]:INTEGER;
125 x1      horizPot[1]:CARDINAL;
126 0B      vertPot[2]:CARDINAL;
127 0T      horizBody[3]:CARDINAL;
128 1x      vertBody[4]:CARDINAL); CODE -210;
129 uw2     (* gadgetID in M2 und fon.c SHORT, in 1.3 dagegen USHORT
*)
130 1Y      (* restlichen Par. in M2 und 1.3 und fon.c USHORT *)
131 V00 PROCEDURE ShowRequester(leftEdge[0]:INTEGER;
132 H90      topEdge[1]:INTEGER;
133 Wb      width[2]:INTEGER;
134 b8      height[3]:INTEGER;
135 T1      backFill[4]:UByte):RequesterPtr; CO
DE -216;
136 A14     (* if fail then return NIL else return RequesterPtr *)
137 w2      (* In 1.3 und M2 backFill : UByte; in fon.c SHORT! *)
138 q80 PROCEDURE RequesterEnd(requesterptr[8]:RequesterPtr); CODE
-222;
139 YL      (* ----- Nachrichtenempfang -----
-----*)
140 RA PROCEDURE Message(windowptr[8]:WindowPtr):INTEGER; CODE -22
8;
141 TW2     (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
Typ CARDINAL sein *)
142 X10 PROCEDURE Class():IDCMPFlagSet; CODE -234;
143 JF PROCEDURE Code():CARDINAL; CODE -240;
144 Ge2     (* fail -> return 0, sonst IntuiMessage.class bzw .code
*)
145 i00 PROCEDURE GadgetID() INTEGER; CODE -246;
146 Pk2     (* fail -> return 0, sonst Gadget.gadgetID *)
147 LW      (* in M2 INTEGER, in 1.3 aber USHORT, siehe Hinweis unten
1*)
148 qy0 PROCEDURE MenuID():CARDINAL; CODE -252;
149 MY2     (* fail -> return 0, sonst die niederwertigsten 5 Bits v
on IntuiMessage.code =
Nummer des Menüs *)
151 8a0 PROCEDURE ItemID():CARDINAL; CODE -258;
152 1E2     (* fail -> return 0, sonst die mittleren 6 Bits von Intu
iMessage.code =
Nummer des Subitems *)
153 195     (* fail -> return 0, sonst die höchstwertigen 5 Bits von
IntuiMessage.code =
Nummer des Subitems *)
154 140 PROCEDURE SubID() CARDINAL; CODE -264,
155 2m2     (* fail -> return 0, sonst die höchstwertigen 5 Bits von
IntuiMessage.code =
Nummer des Subitems *)
156 Jb5     (* fail -> return 0, sonst die höchstwertigen 5 Bits von
IntuiMessage.code =
Nummer des Subitems *)
157 y30 END M2Extintui.
158 15      (* Hinweis: Die Schnittstellen-Module können jederzeit vom
Anwender auf den
neuesten Stand gebracht und kompiliert werden. Als Refer
enz gilt 1.3Include.
Sowohl in fon.c als auch in M2 ist Gadget.gadgetID vom T
yp SHORT(=INTEGER).
In 1.3 jedoch ist Gadget.gadgetID vom Typ USHORT(=CARDIN
AL). *)
(C) 1989 M&T

```

Listing 6. (Schluß)

Meister der Tasten

Sind Sie in der Lage, Ihre Amiga-Tastatur blind zu bedienen? Tippen Sie schnell und fehlerlos? Wenn ja, können Sie die nächsten Zeilen getrost überlesen. Enthaltene Ihre Textypte jedoch jede Menge Tipfehler, dann ist »Keyboard-Master« genau das Richtige für Sie.

Von Robert Kretzschmar

Besonders wenn Sie oft Briefe mit der Schreibmaschine oder einer Textverarbeitung verfassen oder auch hin und wieder Listings abtippen, ist es von großem Vorteil, blind schreiben zu können. Das erhöht nicht nur die Schreibgeschwindigkeit enorm, sondern hilft auch, Tippfehler zu vermeiden. Das Programm »Keyboard-Master« ermöglicht Ihnen das Erlernen des Zehn-Finger-Systems. Es stellt einen vollständigen, interaktiven Schreibmaschinenkurs dar, der in 20 Kapitel unterteilt ist. Die Spanne der Übungen reicht von einfachen »ASDF«-Aufgaben bis hin zu speziellen Fingerübungen für Programmierer.

Um den gewünschten Erfolg zu haben, ist es notwendig, möglichst regelmäßig zu trainieren und anfängliche Probleme durch Wiederholung der Übungen zu beseitigen. Wenn Sie das beachten, wird Ihnen dieses Lernprogramm sicherlich schnell die ersten Erfolgserlebnisse beschern.

Die Lernerfolge mit »Keyboard-Master« sind wirklich erstaunlich. So mancher unserer Redakteure hat sich die Lektionen »reingezogen« und seine »Tipfelerien« enorm reduzieren können. Wer seine Kenntnisse im Maschineschreiben verbessern will, sollte sich das Programm unbedingt ansehen.

»Keyboard-Master« wurde in Amiga-Basic verfaßt. Trotzdem ist die Arbeitsgeschwindigkeit sehr hoch.

Das Listing

»Keyboard-Master« finden Sie auf der Programmservice-Diskette zu diesem Sonderheft. Ein Abdruck des Listings erschien uns sinnig: Erstens ist das Programm sehr lang, und daher zum Abtippen kaum geeignet. Ein weiterer Grund sind die Übungsdateien. Wie gesagt, besteht der Kurs aus zwanzig Kapiteln mit verschiedenen Übungen. Diese müßten also, damit das Programm überhaupt lauffähig ist, eben-

falls eingegeben werden. Da dies sehr mühsam wäre, haben wir uns also entschlossen, »Keyboard-Master« im Heft nicht abzudrucken.

Sie finden also auf der Diskette zu diesem Sonderheft das Basic-Listing unter dem Namen »KeyboardMaster« zusammen mit den Files »Lektion 1.1« bis »Lektion 20.1« und der Übungsdatei »test.k«. Kopieren Sie alle Dateien auf eine Diskette, auf der sich Amiga-Basic befindet. Vor dem Start muß darauf geachtet werden, daß der deutsche Zeichensatz aktiviert ist (vom CLI aus mit »system/setmap d« aktivieren).

Die beiden Zeilen nach dem »CLEAR«-Befehl am Listing-Beginn enthalten zwei »Default«-Werte, die Sie Ihren Ge-

pen wollen, drücken Sie einfach die RETURN-Taste und der von Ihnen im Programmtext voreingestellte Name wird verwendet. Als Filename wäre zum Beispiel Ihr Vorname sinnvoll. Das Programm hängt übrigens selbstständig an den eingegebenen Namen »k.« an, um es als Kursdatei von anderen Files zu unterscheiden. Beim ersten Start des Programms wird dann die Kursdatei mit dem entsprechenden Namen auf Wunsch auf Diskette erzeugt. Auf der Diskette befindet sich übrigens bereits eine Kursdatei unter dem Namen »test.k«. Wenn Sie das Listing entsprechend angepaßt haben, speichern Sie es erneut. Die Arbeit mit dem Programm kann nun beginnen.

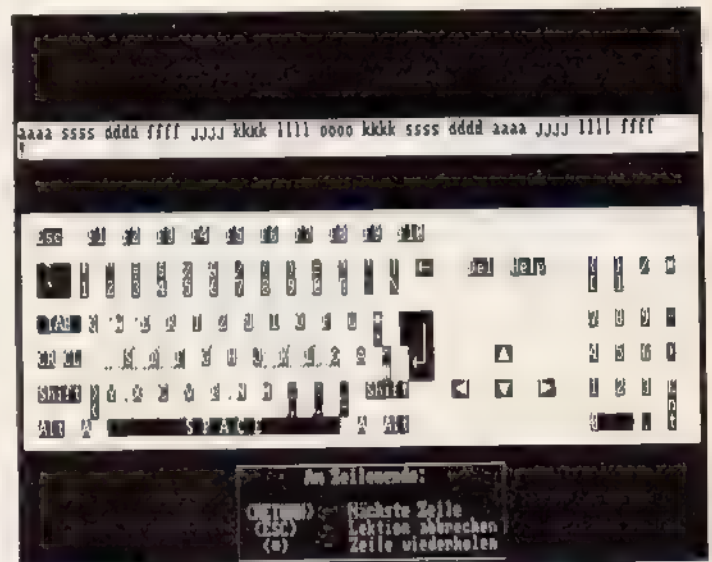


Bild 2. Die erste Übung hat begonnen

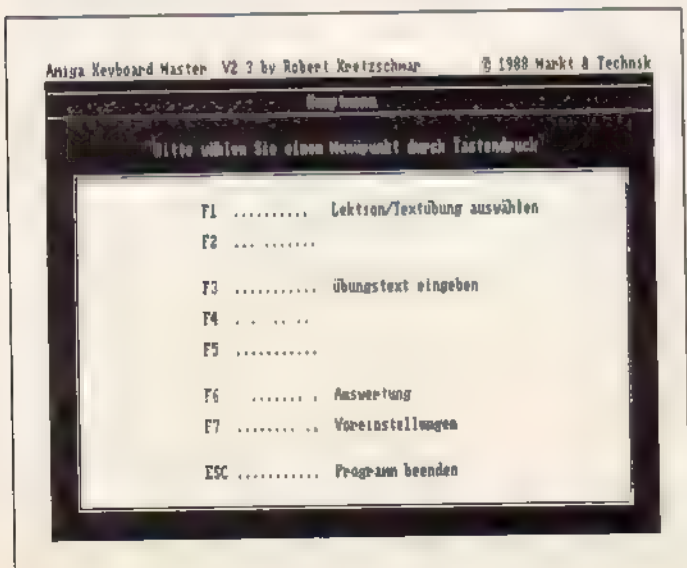


Bild 1. Das Hauptmenü von »Keyboard-Master«

gebenheiten anpassen können: »drive\$« stellt die Laufwerksbezeichnung dar, die immer dann benutzt wird, wenn beim Laden der Kursdatei kein Laufwerk gesondert angegeben wird. Sinnvoll wäre hier, das Laufwerk (»df0:« oder »df1:«) anzugeben, von dem Sie gewöhnlich das Programm laden werden. Direkt darunter steht der voreingestellte Name der Kursdatei. In dieser Datei, die das Programm für jeden Benutzer gesondert anlegt, stehen die Informationen über die bearbeiteten Lektionen und die erzielten Ergebnisse. Der Name dieser Datei wird beim Programmstart immer erfragt. Wenn Sie nicht jedesmal denselben Filenamen eintip-

Die ersten Schritte

Nach dem Start mit »RUN« werden Sie zuerst nach dem Namen der Kursdatei gefragt (siehe oben). Wollen Sie den vorgegebenen Namen verwenden, drücken Sie nur <RETURN>. Anschließend wird die Kursdatei geladen. Wenn sie unter diesem Namen noch nicht existiert, werden Sie gefragt, ob eine neue angelegt werden soll. Beantworten Sie mit »n«, können Sie den Namen nochmals eingeben. Tippen Sie »j« (was Sie beim ersten Start tun sollten), legt das Programm eine neue Datei an. Falls Sie eine bereits existie-

rende Kursdatei überschreiben wollen, müssen vor dem eigentlichen Filenamen drei Sterne und ein Leerzeichen eingegeben werden. Achtung! Alle Daten in dieser Datei gehen dabei verloren! Wichtig ist, daß Sie die Diskette mit der Kursdatei immer im Laufwerk lassen, solange Sie mit dem Programm arbeiten.

Sie befinden sich nun im Hauptmenü (Bild 1). Dabei wird Ihnen auffallen, daß das ganze Programm über die Tastatur gesteuert wird. Schließlich soll deren Bedienung ja erlernt werden. Zudem kann die Bedienung auch ohne Maus komfortabel und einfach sein.

Einige Menüpunkte, die im Moment noch nicht sinnvoll sind, fehlen und sind nicht wählbar. Im folgenden werden alle Menüpunkte der Reihe nach besprochen. Beginnen Sie jedoch nicht sofort, den Kurs durchzuarbeiten. Das sollten Sie erst dann tun, wenn Sie auch die Hinweise zur Durchführung des Kurses gelesen haben. Dies soll Sie aber nicht daran hindern, die Funktion des Programms beim Lesen der Bedienungsanleitung auszuprobieren.

<F1> - Lektion/Textübung wählen

Mit der Funktionstaste <F1> gelangen Sie vom Hauptmenü in ein Untermenü, in dem alle 20 Lektionen aufgelistet sind. Wählen Sie eine davon aus, indem Sie die jeweilige Zahl, gefolgt von <RETURN>, eingeben. Im Anschluß daran werden alle Lektionen in den Speicher geladen. Wenn Sie später weitere Lektionen anwählen, erfolgen keine weiteren Zugriffe auf die Diskette. Wenn Sie keine reguläre Lektion bearbeiten wollen, sondern einen beliebigen ASCII-Text üben wollen, drücken Sie statt einer Zahl nur die RETURN-Taste und Sie werden nach dem Filenamen des Textes gefragt. Tippen Sie den Namen des gewünschten Files ein. Nach dem Ladevorgang gelangen Sie wieder ins Haupt-Menü.

<F2> - Lektion starten

Je nachdem, ob Sie eine Lektion oder einen Übungstext geladen haben, hat dieser Menüpunkt den Titel »Lektion xx starten« oder »Textübung starten«. Nach Betätigen von <F2> werden zunächst einige Daten aus der Kursdatei geladen und der wichtigste Teil des Programms beginnt.

Auf dem Bildschirm wird ein Abbild einer Amiga-Tastatur, die an jene des Amiga 500 an-

gelehnt ist, gezeichnet (Bild 2). Die Querlinien mitten im Tastenfeld sollen am Anfang helfen, alle Tasten mit dem richtigen Finger anzuschlagen. Die Hilfslinien trennen also den »Zuständigkeitsbereich« der einzelnen Finger voneinander. Die helle Linie in der Mitte stellt die Trennung zwischen rechter und linker Hand dar. Wenn in den Voreinstellungen (<F7>) das Zeichnen der Hilfslinien nicht angeschaltet ist, sind diese natürlich nicht zu sehen. Genauso verhält es sich mit der Hilfstastatur und der Anzeige der jeweils zu drückenden Taste (siehe unten).

Der Bildschirmaufbau benötigt einige Sekunden, jedoch nur beim ersten Aufruf dieses Programmteils. Danach ist die Grafik im RAM gespeichert. Ganz oben sehen Sie die Statuszeile, in der Sie lesen können, welche Lektion Sie gerade bearbeiten, welche Nummer die aktuelle Zeile hat und in welchem Teil der Lektion (Griff-, Wort- oder Satzübung) Sie sich im Moment befinden. Darunter wird die Anzahl der Fehler angezeigt, die Sie in der aktuellen Zeile gemacht haben. Noch weiter unten sehen Sie in einem farbigen Feld den Klartext der jeweiligen Zeile. Unter dieser befindet sich ein nach oben zeigender Pfeil, der auf das Zeichen zeigt, das eingetippt werden muß.

Im unteren Teil des Bildschirms befindet sich noch ein gelber Kasten, der als Hilfe für den Benutzer die möglichen Tasten am Zeilenende beinhaltet. Ist der Bildschirmaufbau abgeschlossen, ertönt ein Summen, das zeigt, daß es losgehen kann. Sie müssen nun Buchstaben für Buchstaben aus der angezeigten Textzeile abtippen. Der Pfeil wandert dabei von Zeichen zu Zeichen mit. Um Sie davon abzuhalten, spähende Blicke auf die Computertastatur zu werfen, werden die zu drückenden Tasten auf der Hilfstastatur hervorgehoben (sofern in der Voreinstellung diese Funktion aktiviert wurde).

Machen Sie einen Fehler, hören Sie einen Piepston und der Pfeil verwandelt sich unter dem falsch getippten Zeichen in ein Ausrufungszeichen (!). Die Fehleranzeige wird um eins erhöht. Tippen Sie nun das Zeichen richtig, bekommt der Cursor (Pfeil) wieder sein normales Aussehen und Sie können weitertippen. Am Zeilenende angelangt erscheint an Stelle des Cursors ein Linkspfeil «-» und Sie können

nun die RETURN-Taste drücken, um in die nächste Zeile zu gelangen, die ESC-Taste, um die Lektion ohne Speichern der Ergebnisse abzubrechen oder <+>, um die getippte Zeile nochmals einzugeben. In der Gesamtwertung werden dann nur jene Fehler, die beim weiteren Durcharbeiten der Zeile gemacht wurden, berücksichtigt.

Um die Funktionsweise auszutesten und sich daran zu gewöhnen, sollten Sie einmal eine Zeile abtippen, am Zeilenende aber nicht die RETURN-Taste drücken, sondern <ESC>. Schließlich handelte es sich nur um einen Versuch, und ein Speichern der Ergebnisse ist wohl kaum erwünscht.

Der Computer weiß alles...

Am Ende jeder Lektion werden die erzielten Ergebnisse, sowie allgemeine Angaben zur Lektion, am Bildschirm angezeigt (Bild 3) und in die Kursdatei gespeichert. So weiß das Programm immer, welche Lektion Sie schon durchgearbeitet haben, wieviel Fehler Sie dabei gemacht haben und wie schnell Sie beim Tippen waren. Genauso ist der Computer — und damit auch Sie — ständig im Bilde darüber, wieviel prozentual falsch gemacht wurde und wie viele Anschläge pro Minute Sie erreicht haben. Schließlich ist er auch noch darüber informiert, ob und wie oft Sie einzelne Lektionen wiederholt und sich dabei verbessert oder verschlechtert haben. Erwähnenswert ist hierbei auch die grafische Anzeige des Fehlerprozentsatzes. All das schauen Sie sich am besten erst an, wenn Sie mit dem eigentlichen Kurs ernsthaft beginnen wollen, denn die Bedienung dieses Programmteils erklärt sich von selbst.

<F3> - Übungstext eingeben

Dieses Lernprogramm gibt Ihnen nicht nur die Möglichkeit, fertige Lektionen zu bearbeiten, es können vielmehr auch eigene Texte eingegeben werden, die Sie zur Übung abtippen können. Dieser Programmteil ist ebenfalls weitgehend selbsterklärend. Es bleibt nur zu bemerken, daß Sie die Eingabe der Zeilen (maximal 20 Zeilen, je Zeile höchstens 78 Zeichen) mit einer Leerzeile abschließen müssen, um ins Hauptmenü zu gelangen. Dort werden Sie bemerken, daß unter »F2« nun »Textübung star-

ten« angeführt ist. Schließlich wollen Sie jetzt Ihren Text bearbeiten, nicht die Lektion. Dasselbe gilt beim Laden eines Übungstextes (siehe oben).

<F4> - Lektions-/Übungstext anzeigen

Über diesen Menüpunkt wird der aktuelle Lektionstext beziehungsweise der eingegebene oder geladene Übungstext angezeigt. Dabei fallen hier die violetten Zahlen am Zeilenanfang auf. Diese zeigen an, wie oft jede Zeile abgetippt werden muß.

Bei der Eingabe eines Übungstextes wird übrigens automatisch eine »1« vor jede Zeile geschrieben. Auch wenn Sie eine ASCII-Datei laden (Menüpunkt 1), wird jede Zeile, die nicht mit einer Ziffer beginnt, mit einer »1« am Zeilenanfang ergänzt. Bei den Lektionstexten können Sie mit einem ASCII-Editor die Anzahl der Wiederholungen leicht ändern. Beachten Sie jedoch, daß nur die erste Ziffer als Wiederholungszahl erkannt wird. Diesen Programmteil verlassen Sie, indem Sie eine beliebige Taste betätigen.

<F5> - Übungstext speichern

Dieser Menüpunkt kann nur dann gewählt werden, wenn Sie zuvor einen Übungstext geladen oder eingegeben haben. Das Programm fragt Sie nach dem gewünschten Filenamen. Wollen Sie den Text doch nicht speichern, drücken Sie nur <RETURN>.

<F6> - Auswertung

In diesem Programmteil werden die Daten, die die Kursdatei über die einzelnen Lektionen enthält, angezeigt (Bild 4). Lektionen, die noch nicht bearbeitet worden sind, werden dabei dunkel dargestellt. Die ausgewählte Lektion wird besonders gekennzeichnet. Das Feld »Differenz FQ« enthält, wenn die Lektion mehr als einmal bearbeitet worden ist, die Differenz des neueren Fehlerquotienten (FQ) zum älteren. Hier sehen Sie, ob Sie sich beim Wiederholen verbessert oder verschlechtert haben. Direkt daneben befindet sich eine weitere Spalte, die Ihnen noch unbekannt ist: Unter »Punkte« wird für jede Lektion eine bestimmte Zahl angegeben, die etwa mit einer Note vergleichbar ist. Bei Punktwerten unter 50 werden dahinter drei Ausrufungszeichen geschrieben, was bedeutet, daß Sie diese Lektion unbedingt wiederholen sollten. Werte über 100 sind sehr gut. In die Punktwertung

Aniga Keyboard Master V2.3 by Robert Kretzschmar © 1988 Markt & Technik	
ENDE von Lektion 1	
Ergebnis	
Anschläge:	662
Fehler:	49
Fehlerquote:	7.41 %
benötigte Zeit:	9 Minuten, 35 Sekunden
Anschläge pro Minute:	69
Niederholung der Lektion:	keine
Drücken Sie eine Taste	

Bild 3. Die Auswertung gibt Aufschluß über den Lernerfolg und Ihren momentanen Leistungsstand

geht neben einigen anderen Faktoren hauptsächlich der Fehlerquotient ein.

<F7> - Voreinstellungen

Mit <F7> gelangen Sie in den Programmteil, in dem Sie verschiedene Voreinstellungen (vergleichbar mit Preferences der Workbench) ändern können. Diese Änderungen werden in die Kursdatei geschrieben und bei jedem erneuten Laden der Kursdatei im Programm berücksichtigt.

Sie haben sechs Möglichkeiten, Einstellungen und damit den Programmablauf zu verändern. Es handelt sich dabei um »Schalter«, wobei durch Druck auf die links neben der jeweiligen Bezeichnung stehende Zahlentaste zwischen zwei Stellungen hin- und hergeschaltet werden kann. Bei einigen Schalterkombinationen nimmt das Programm selbstständig Änderungen der Stellungen vor, das heißt es verhindert unlogische Kombinationen. Kommen wir nun zur Bedeutung der einzelnen Schalter:

Wenn Schalter 1 ausgeschaltet ist, wird die Hilfstastatur nicht mehr angezeigt.

Schalter 2 ist für die Hervorhebung der jeweils zu drückenden Tasten auf der gezeichneten Hilfstastatur zuständig.

Mit Schalter 3 wird bestimmt, ob die Textzeile immer angezeigt wird oder nicht. Es ist sinnvoll, diese auszuschalten, wenn man sich voll auf die Griffe und die Hilfstastatur konzentrieren möchte.

Ist Schalter 4 ausgeschaltet, wird das Zeichnen der Hilfslinien auf der Tastatur unterdrückt, da diese mit der Zeit störend wirken können.

Wenn Sie das automatische Wiederholen einzelner Zeilen unterdrücken wollen, müssen Sie Schalter 5 ausschalten. Dies ist dann sinnvoll, wenn Sie eine Lektion schon oft bearbeitet haben und diese nur kurz auffrischen wollen.

Mit dem sechsten und letzten Schalter wählen Sie schließlich das Default-Laufwerk (df0: oder df1:), das immer dann verwendet wird, wenn bei Filenamen keine Laufwerksangabe gesondert angegeben wird. Auch die Lektionstexte werden stets von diesem Laufwerk geladen. Bitte verwechseln Sie diese Einstellung nicht mit dem Default-Laufwerk, das Sie vor dem ersten Programmstart im Listing direkt eingestellt haben. Dieses Laufwerk gilt nur für die Kursdatei, ansonsten hat das in diesem Programmteil einzustellende Laufwerk Priorität.

Mit <ESC> werden eventuelle Änderungen in die Kursdatei gespeichert und Sie kommen zurück ins Hauptmenü.

ESC - Programm beenden

Mit <ESC> im Haupt-Menü verlassen Sie das Programm.

Der Kurs beginnt

Ich hoffe, Sie haben nun die ersten Gehversuche mit Keyboard-Master hinter sich und brennen darauf, endlich den Kurs zu beginnen.

Dazu greifen Sie zunächst, ohne das Programm zu starten, die Grundstellung, die Sie in Tabelle 1 (Seite 35) finden. Diese Fingerstellung ist sehr wichtig, prägen Sie sich diese ein und versuchen Sie mehrmals, diese zu finden, ohne auf die Tastatur zu sehen.

Aniga Keyboard Master V2.3 by Robert Kretzschmar © 1988 Markt & Technik					
Auswertung (Kursdatei df0:test)					
Lektion	Niederh.	Fehlerquotient	Anschl./min	Differenz FQ	Punkte
1	9 mal	7.41 %	69		98
2	0 mal	5.47 %	177		112
3	0 mal	3.81 %	150		111
4	0 mal	6.58 %	118		97
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
Taste drücken					

Bild 4. Durch das Anlegen einer Kursdatei auf Diskette wissen Sie ständig über Ihren »Wissens-Status« beschied

Sie sollten nun das Programm erneut starten und einen beliebigen Filenamen für die Kursdatei (etwa Ihren Vornamen) eingeben. Wenn Sie bei den ersten Versuchen schon eine Datei unter diesem Namen erstellt haben, geben Sie drei Sterne und ein Leerzeichen vor dem eigentlichen Namen ein, so daß in jedem Fall eine völlig neue Kursdatei angelegt wird. Nun sehen Sie sich die aktuellen Voreinstellungen des Programms mit <F7> an. Ändern sollten Sie zunächst allerdings nur das Default-Laufwerk, um es Ihren Gegebenheiten anzupassen. Gehen Sie zurück ins Hauptmenü und drücken Sie <F1>. Dort geben Sie »1« gefolgt von <RETURN> ein. Nach dem Ladevorgang können Sie die Lektion starten. Bitte sehen Sie sich aber vorher die folgenden zehn Grundregeln (das sollten Sie zur besseren Einprägung möglichst vor jedem Start einer Lektion tun) an:

1. Arbeiten Sie möglichst regelmäßig mit dem Programm.
2. Bearbeiten Sie an einem Tag nur eine Lektion.
3. Wiederholen Sie jede Lektion am besten mindestens einmal.
4. Lassen Sie sich nicht von anfänglichen Schwierigkeiten entmutigen.
5. Greifen Sie die Grundstellung jedesmal blind.
6. Sehen Sie nie auf die Tastatur, orientieren Sie sich an der Hilfstastatur am Bildschirm.
7. Der Anschlag erfolgt kurz und ruckartig aus dem Muskel heraus.
8. Gehen Sie nach jedem Anschlag schnell in die Grundstellung zurück.

9. Tippen Sie langsam, aber gleichmäßig im Takt.

10. Halten Sie die Handgelenke möglichst hoch. Die Hand darf nicht flach auf der Tastatur liegen, sie muß angewinkelt sein.

11. Es kann nicht schaden, wenn Sie Ihre Finger mit einer speziellen Gymnastik trainieren.

Noch ein Hinweis zu Lektion 12: Beachten Sie, daß eine zu »shiftende« Taste, die sich im rechten Teil der Tastatur befindet, immer zusammen mit der linken SHIFT-Taste (und umgekehrt) angeschlagen wird. Das Programm unterstützt Sie dabei, indem immer die richtige zu drückende SHIFT-Taste hervorgehoben wird.

Haben Sie sich all dies zu Herzen genommen, kann es losgehen: Drücken Sie <F2> und tippen Sie nun alle Zeilen der Lektion ein. Schließen Sie jede Zeile mit <RETURN> ab. Werden Sie nicht bei jedem Fehler nervös. Errare humanum est! (lat.: Irren ist menschlich!) Bleiben Sie konzentriert und denken Sie daran, daß die ersten Lektionen die schwierigsten sind.

Sind Sie mit der Lektion fertig, werden Ihnen die Ergebnisse gezeigt. Hinter Fehler, Fehlerquotient und Anschläge pro Minute stehen je zwei Zahlen, jedoch nur dann, wenn Sie die Lektion insgesamt mindestens zweimal bearbeitet haben. Die zweite Zahl stellt jeweils die Differenz zum Ergebnis des letzten Mals dar. Die Anzeige der Anschläge je Minute sollten Sie bei den ersten Lektionen einfach übersehen. Sie sind beim Lernen völlig unwichtig und werden in der

Punktwertung kaum berücksichtigt. Wichtig ist, daß Sie möglichst gleichmäßig tippen und wenige Fehler machen. Wenn Sie nun vom Hauptmenü mit <F6> die Auswertung aufrufen, können Sie sich die erreichte Punktzahl ansehen. Sie sollten diese jedoch nicht mit Schulnoten vergleichen. Denken Sie daran, daß Sie hier völlig ohne Druck lernen. Die Punkte und auch die

Tippgeschwindigkeit sollen Ihnen vor allem als Vergleich dienen, wenn Sie schon einige Lektionen hinter sich haben. Punktwerte unter »50« sind jedoch ein Warnzeichen dafür, daß Sie diese Lektion wiederholen müssen. Wenn Sie mehr als »80« Punkte erreichen, können Sie sehr zufrieden sein, Werte über »100« oder gar über »120« sind sehr gut.

M. Jobst/kn

Linke Hand:	a	kleiner Finger
	s	Ringfinger
	d	Mittelfinger
	f	Zeigefinger
Rechte Hand:	j	Zeigefinger
	k	Mittelfinger
	l	Ringfinger
	ö	kleiner Finger

Tabelle 1. Die Grundstellung beim Maschineschreiben ist die Basis für alle Übungen

Sounds im Eigenbau

Von Arno Gölzer

Wenn auf dem Mond riesige Meteore einschlagen und dabei gewaltige Krater hinterlassen, so geschieht das vollkommen lautlos. Schuld daran ist die fehlende Lufthülle. Der Schall ist eine hörbare oder zumindest nachweisbare Schwingung, die sich nur in einem Medium, beispielsweise in der Luft oder im Wasser, verbreitet. Jedes Geräusch hat seine eigene, individuelle Schwingung und kann daher eindeutig identifiziert werden. Aber auch eine — zumindest annähernde — mathematische Beschreibung ist möglich. Wir unterscheiden zwischen vier Grundschwingungen:

- Sinus
- Rechteck
- Dreieck
- Sägezahn

Die Klangfarbe eines Tones machen aber im wesentlichen die sogenannten Oberschwingungen aus. Es handelt sich um mehr oder weniger große Abweichungen von der eigentlichen Wellenform. Der Amiga läßt uns in bezug auf das Aussehen der Wellenform völlig freie Hand. Somit ist es möglich, beliebige Geräusche künstlich zu erzeugen — wir müssen nur die Wellenform entsprechend beschreiben. Diese

donner in Spielen reicht das Spektrum. Der Clou: Die Sounds werden als Quellcode gespeichert und sind so leicht für eigene Programme zu verwenden.

Aufgabe wird mit dem Sound-Effekte-Editor (SEEd) (Listing 1) zum Kinderspiel.

Nach dem Eingeben des Programms mit dem Checksummer (Seite 159) und dem Compilieren starten Sie SEEd vom CLI aus, indem Sie den Programmnamen eingeben und <RETURN> drücken. Haben Sie ein Programm-Icon für SEEd entworfen, können Sie den Soundgenerator auch von der Workbench aus starten. Wichtig ist in jedem Fall, daß sich im Verzeichnis »Libs« die Mathematik-Libraries befinden (mathtrans.library)

Von der Welle zum Klang

Nach dem Programmstart öffnet sich ein Fenster, das in zwei Felder unterteilt ist (Bild 1):

1. Das Bedienungsfeld belegt das rechte Viertel des Bildschirms. Es besteht aus insgesamt zwölf Reglern (Gadgets). Lesen Sie im folgenden deren Funktion:

Mit dem »Sound-Effekte-Editor« erzeugen Sie komfortabel alle Geräusche. Vom einfachen akustischen Hinweis auf Eingabefehler bis zum Kanonen-

Großer Schieberegler rechts: Feineinstellen der Frequenz. Mit den beiden Frequenzreglern können Sie über 65000 Frequenzen einstellen!

Kleiner Schieberegler links: Hier stellen Sie ein, wie oft der eingestellte Klang wiederholt werden soll.

Kleiner Schieberegler rechts: Die Lautstärke.

Null: Wellenform auf Null stellen (waagerechter Strich erscheint).

Großer Schieberegler links: Grobeinstellen der Frequenz. Schieben Sie den Reglerknopf nach unten, wenn Sie niedrigere Frequenzen einstellen möchten.

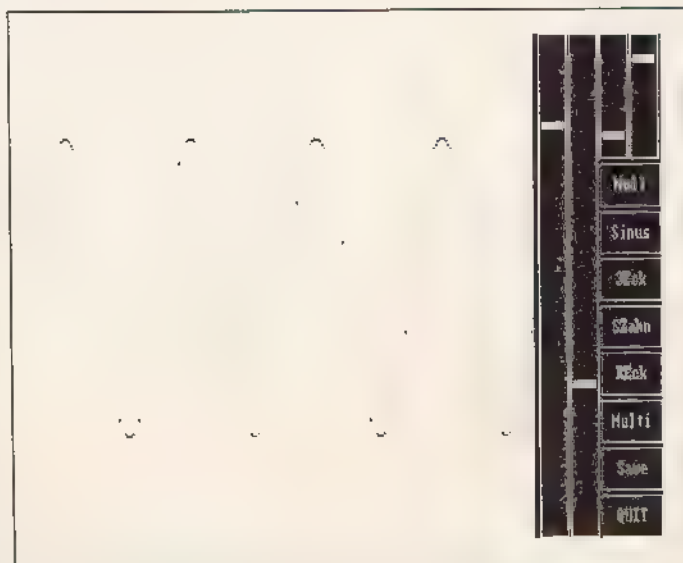


Bild 1. Das SEEd-Fenster. Im linken Teil bearbeiten Sie den gewünschten Sound, rechts befinden sich alle Regler.

Sinus:

Einstellen der Wellenform auf Sinus.

3Eck:

Eine Dreieckschwingung wird eingestellt.

SZahn:

Vorwahl der Sägezahn-schwingung.

REck:

Die Rechteckschwingung einstellen.

Die voreingestellten Schwin-gungen können jetzt beliebig verändert werden (siehe unten). Allerdings ist in diesem Zu-stand der Klang nicht immer op-timal. Verbessern Sie die Quali-tät, indem Sie mehrere Schwin-gungen im Zeichenfeld unter-bringen:

Multi:

Eingestellte Schwingung ver-doppeln.

Save:

»Geräusch« auf Diskette si-chern. Ein Requester öffnet sich. Geben Sie den Filenamen an und klicken Sie das Gadget mit der Aufschrift »Save« an (oder »Break«, wenn Sie lieber nicht speichern möchten).

Quit:

Mit diesem Schalter brechen Sie das Programm ab.

Im weiß hinterlegten Zei-chenfeld können Sie vorgeferti-gte Schwingungen verändern

Sounds zeichnen mit der Maus

oder aber völlig neue entwer-fen. Drücken Sie die linke Maustaste und »zeichnen« Sie Ihren Soundeffekt. Die besten Ergebnisse erhalten Sie, wenn Sie die Höhe des Zeichenfeldes möglichst voll ausnutzen. Im anderen Fall sind die Schwin-gungen nur schlecht hörbar.

Weiterhin ist es von Vorteil, wenn Schwingungsanfang und -ende auf gleicher Höhe liegen. Nutzen Sie auch hier die Multi-Funktion zur Verbesserung der Soundqualität. Sie hören den Klang, wenn Sie die rechte Maustaste betätigen. Bei den Grundschnungen ist aller-dings selten ein Ton zu hören. Keine Angst, das Programm funktioniert trotzdem fehlerfrei. Verdoppeln Sie die Grund-schwingung mehrfach, ent-locken Sie dem Amiga bereits einen Klang.

Begrenzen der Wellenform

Mit den orangefarbenen Mar-kierungen im Zeichenfeld kön-nen Sie den Anfang und das Ende der Wellenform selbst festlegen. Bedienen Sie die

Markierungen mit dem Ziffern-Block der Tastatur:

<7> Startmarkierung nach links,
<9> Startmarkierung nach rechts,
<8> Startmarkierung auf Start

Da das Datenfeld auf einem Langwort (Longword) beginnen muß, erfolgt die Bewegung der Startmarkierung in 4er-Sprüngen.

<4> Endemarkierung nach links,

Einbindung in C-Programme

<6> Endemarkierung nach rechts,
<5> Endemarkierung auf Ende.

Das Ende des Datenfeldes ist nicht an bestimmte Speicher-bereiche gebunden. Daher ist hier noch eine Feineinstellung in Einerschritten möglich:

<1> Endemarkierung einen Schritt nach links,
<3> Endemarkierung einen Schritt nach rechts,
<2> Endemarkierung auf Ende.

Das Einbinden eines SEED-Sounds in C-Programme ist sehr einfach, da SEED den Sound als C-Quellcode ablegt. Folgende Schritte sind nötig:

- Binden Sie die SEED-Datei als Headerfile in Ihr eigenes Programm ein.

- Rufen Sie zu Beginn Ihres ei-genen Programms die Funktion SoundOn() auf. Die Funktion liefert im Fehlerfall den Wert 1.

Ihr Sound ist nun initialisiert und Sie können ihn jederzeit und beliebig oft abspielen:

- »Play()« spielt den Soundef-fekt ab.

- Rufen Sie vor Beendigung Ihres Programms die Funktion »SoundOff()« auf. Sie gibt alle für den Soundeffekt notwandi-gen Ressourcen wieder frei.

Play() und SoundOff() liefern keinen Wert.

Hier ein kleines Beispiel:

```
#include "SoundName"

main()
{
    if(!SoundOn()){
        Play();
        SoundOff();
    }
}
```

Aztek-C-Besitzer übersetzen dieses Beispiel so:

```
cc NAME.c +L
ln NAME.o -lc32
```

Anstelle von »NAME« muß natürlich der Name Ihres Pro-

gramms stehen. Programmie- rer anderer Programmierspra- chen profitieren immer noch von den gesicherten Soundda- ten. Deshalb enthält Listing 2 für den Interessierten zum Schluß noch das dokumentier- te Listing einer SEED-Sound- datei. Mit dem SEED geben wir

Ihnen ein wertvolles Werkzeug zur Erzeugung von Sounds in eigenen Programmen in die Hand. Ein Werkzeug, das die Soundprogrammierung unge- mein erleichtert und das Sie be- stimmt bald nicht mehr missen wollen.

Arno Gölzer/kn

Programmname:	SEED
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztek V3.4
Aufrufe:	cc SEED.c +L ln SEED.o -lm32 -lc32
Bemerkung:	benötigt Math.-Libraries

Programmautor Arno Gölzer

```
1 #00 /*
2 pt **      Gölzer's SEED.c
3 PR **      08.88
4 7% */
5 #f #include <math.h>
6 #F #include <stdio.h>
7 #A #include <exec/types.h>
8 #l #include <intuition/intuition.h>
9 #e #include <graphics/gfx.h>
10 #Y #include <devices/audio.h>
11 #I #include <exec/memory.h>
12 #K #define GH (0xFFFF/255)
13 #t #define GL (0xFFFF/255)
14 #C #define GC (0xFFFF/64)
15 #p #define GV (0xFFFF/64)
16 #G #define ENDE (((Ende-Start)/2)+1)
17 #L #define MOUSEX (win->MouseX)
18 #U #define MOUSEY (win->MouseY)
19 #F #define GAD_ID (((struct Gadget *)ADDR)->GadgetID)
20 #a #PTR ADDR;
21 #L ULONG CLASS, CODE;
22 #Y UBYTE *Data,buffer=15,name[31],undo[31],Title[]={(UBYTE *)"-
  *- Gölzer's SEED -*-"};
23 #M UWORD l,h,p,c,v;
24 #A struct Image HDummy,LDummy,CDummy,VDummy;
25 #A SHORT Start=0,Ende=510;
26 #S SHORT OKpairs[] = { 0,0, 61,0, 61,21, 0,21, 0,0 } ;
27 #f SHORT Reqpairs[] = { 0,0, 246,0, 246,67, 0,67, 0,0 } ;
28 #y SHORT Spairs[] = { 0,0, 210,0, 210,11, 0,11, 0,0 } ;
29 #r struct Border OKRand = { -1, -1, 1,0, JAM1, 5,&OKpairs[0],
  NULL } ;
30 #S struct Border ReqR = { 2, 1, 1,0, JAM1, 5,&Reqpairs[0],
  NULL } ;
31 #P struct Border SRand = { -1, -2, 1,0, JAM1, 5,&Spairs[0],
  NULL } ;
32 #p struct IntuiText OKT2 = { 1,0,JAM1,14,6,NULL,(UBYTE *) "QUIT
  ",NULL } ;
33 #l struct IntuiText OKText={ 3,0,JAM1,15,7,NULL,(UBYTE *) "QUIT
  ",&OKT2 } ;
34 #N struct IntuiText SaText={ 1,0,JAM1,14,6,NULL,(UBYTE *) "Save
  ",NULL } ;
35 #W struct IntuiText MText = { 1,0,JAM1,10,6,NULL,(UBYTE *) "Mult
  1",NULL } ;
36 #h struct IntuiText RText = { 1,0,JAM1,14,6,NULL,(UBYTE *) "REck
  ",NULL } ;
37 #G struct IntuiText ZText = { 1,0,JAM1,10,6,NULL,(UBYTE *) "SZab
  n",NULL } ;
38 #E struct IntuiText DText = { 1,0,JAM1,14,6,NULL,(UBYTE *) "3Eck
  ",NULL } ;
39 #A struct IntuiText SText = { 1,0,JAM1,10,6,NULL,(UBYTE *) "Sinu
  s",NULL } ;
40 #A struct IntuiText NText = { 1,0,JAM1,14,6,NULL,(UBYTE *) "Null
```

Listing 1. Mit dem Sound-Effekte-Editor (SEED) entwerfen Sie Geräusche mit der Maus

```

    ",NULL    ];
41 aL struct IntuiText ReqT2 =[ 1,0,JAM1,45,6,NULL,Titel,NULL
    ];
42 yx struct IntuiText ReqT  =[ 3,0,JAM1,46,7,NULL,Titel,&ReqT2
    ];
43 i3 struct IntuiText RCaT  =[ 1,0,JAM1,10,6,NULL,(UBYTE *)"Brea
    k",NULL    ];
44 FC struct IntuiText RTGT  =[ 1,0,JAM1,70,-10,NULL,(UBYTE *)"Fil
    ename:",NULL];
45 qZ struct PropInfo PHInfo =[ FREEVERTI AUTOKNOB,0,0,    0,GH,0
    ,0,0,0,0,0 ];
46 V7 struct PropInfo PLInfo =[ FREEVERTI AUTOKNOB,0,0L*255,0,0L,0
    ,0,0,0,0,0 ];
47 Vw struct PropInfo CInfo  =[ FREEVERTI AUTOKNOB,0,0C*64, 0,0C,0
    ,0,0,0,0,0 ];
48 be struct PropInfo VInfo  =[ FREEVERTI AUTOKNOB,0,0,    0,GV,0
    ,0,0,0,0,0 ];
49 9C struct StringInfo SInfo=[ name,undo,0,31,0,0,0,0,0,0,0,NU
    LL        ];
50 uM struct Gadget RTGad    =[ NULL,20,30,210,10,GADGHCOMP,
51 eU4  GADGIMMEDIATEI RELVERIFY,STRGADGETI REQGADGET,
52 wb  (APTR)&SRand,NULL,&RTGT,0,(APTR)&SInfo,14,NULL
    ];
53 6d0 struct Gadget RCa    =[ &RTGad,20,45,60,20,GADGHCOMP,
54 ee4  ENDGADGETI GADGIMMEDIATEI RELVERIFY,BOOLGADGETI REQGADGET,
55 e4  (APTR)&KRand,NULL,&RCaT,0,0,13,NULL
    ];
56 we0 struct Gadget ROK    =[ &RCa,170,45,60,20,GADGHCOMP,
57 rF4  ENDGADGETI GADGIMMEDIATEI RELVERIFY,BOOLGADGETI REQGADGET,
58 W3  (APTR)&KRand,NULL,&SaText,0,0,12,NULL
    ];
59 nu0 struct Gadget OKGad  =[ NULL,579,235,60,20,GADGHCOMP,
60 r84  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
61 JP  (APTR)&KRand,NULL,&OKText,0,0,11,NULL
    ];
62 6d0 struct Gadget SaGad  =[ &OKGad,579,211,60,20,GADGHCOMP,
63 uB4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
64 Ua  (APTR)&KRand,NULL,&SaText,0,0,10,NULL
    ];
65 M00 struct Gadget MGad   =[ &SaGad,579,187,60,20,GADGHCOMP,
66 xE4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
67 TV  (APTR)&KRand,NULL,&MText,0,0,9,NULL
    ];
68 Me0 struct Gadget RGad   =[ &MGad,579,163,60,20,GADGHCOMP,
69 OH4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
70 ek  (APTR)&KRand,NULL,&RText,0,0,8,NULL
    ];
71 fV0 struct Gadget ZGad   =[ &RGad,579,139,60,20,GADGHCOMP,
72 JK4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
73 v8  (APTR)&KRand,NULL,&ZText,0,0,7,NULL
    ];
74 ou0 struct Gadget DGad   =[ &ZGad,579,115,60,20,GADGHCOMP,
75 6N4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
76 E4  (APTR)&KRand,NULL,&DText,0,0,6,NULL
    ];
77 Re0 struct Gadget SGad   =[ &DGad,579,91,60,20,GADGHCOMP,
78 9Q4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
79 jn  (APTR)&KRand,NULL,&SText,0,0,5,NULL
    ];
80 vy0 struct Gadget NGad   =[ &SGad,579,67,60,20,GADGHCOMP,
81 OT4  GADGIMMEDIATEI RELVERIFY,BOOLGADGET,
82 aY  (APTR)&KRand,NULL,&NText,0,0,4,NULL
    ];
83 Ye0 struct Gadget VGad   =[ &NGad,608,0,30,64,0,0,PROPGADGET,
84 e14  (APTR)&VDummy,NULL,NULL,0,(APTR)&VInfo,3,NULL
    ];
85 SD0 struct Gadget CGad   =[ &VGad,577,0,30,64,0,0,PROPGADGET,
86 pO4  (APTR)&CDummy,NULL,NULL,0,(APTR)&CInfo,2,NULL
    ];
87 H30 struct Gadget PGadL  =[ &CGad,546,0,30,256,0,0,PROPGADGET
    ,
88 QH4  (APTR)&LDummy,NULL,NULL,0,(APTR)&PLInfo,1,NULL
    ];
89 u10 struct Gadget PGadH  =[ &PGadL,515,0,30,256,0,0,PROPGADGE
    T,
90 mo4  (APTR)&HDummy,NULL,NULL,0,(APTR)&PHInfo,0,NULL
    ];
91 j10 struct Requester Req=[ NULL,10,10,250,70,0,0,
92 KD4  BROK,&ReqR,&ReqT,0,0,NULL,NULL,NULL,NULL
    ];
93 a00 struct NewWindow BW  =[
94 Gw4  0,0,640,256,-1,-1,VANTILAKEY1 MOUSEBUTTONS1 GADGETUP,

```

```

95 K2  ACTIVATEI BORDERLESSI RMBTRAP,&PGadH,NULL,NULL,NULL,NULL,
96 JJ  0,0,0,0,&BENCHSCREEN
    ];
97 w40 struct Library      *OpenLibrary();
98 Hq struct IntuiMessage  *GetMsg();
99 b1 struct MsgPort       *SPort,*CreatePort();
100 cq struct Device        *SDev,*OpenDevice();
101 vE struct IntuitionBase *IntuitionBase,
102 JD struct GfxBase       *GfxBase;
103 Ws struct MathTransBase *MathTransBase;
104 Wp struct Window        *win,*OpenWindow();
105 Ls struct RastPort      *RP;
106 y0 struct IOAudio       Sound;
107 Bd VOID closew()        /* zum Programmende alle Re
    sourcen freigeben */
108 g9 {
109 584 if(!SDev)      CloseDevice(&Sound ,
110 g0 if(SPort)      DeletePort(SPort,sizeof(struct MsgPor
    t));
111 iu if(Data)      FreeMem(Data,256*sizeof(UBYTE));
112 J if(win)        CloseWindow(win),
113 aS if(MathTransBase) CloseLibrary(MathTransBase),
114 de if(GfxBase)    CloseLibrary(GfxBase);
115 LA if(IntuitionBase) CloseLibrary(IntuitionBase);
116 Jh exit(0);
117 t00 {
118 6H VOID openw()      /* Programmstart: Libraries, Windo
    w, ChipMemory ... */
119 rK {
120 s14 UBYTE *AllocMem();
121 GP if(!IntuitionBase=(struct IntuitionBase *)
    OpenLibrary("Intuition.library",0))) close
    w();
122 BuC if(!GfxBase=(struct GfxBase *)
    OpenLibrary("graphics.library",0))) close
    w();
123 KB4 if(!MathTransBase=(struct MathTransBase *)
    OpenLibrary("mathtrans.library",0))) close
    w();
124 sVC
125 h24 if(!MathTransBase=(struct MathTransBase *)
    OpenLibrary("mathtrans.library",0))) close
    w();
126 foC
127 604 if(!win=OpenWindow(&BW))) close
    w();
128 hE if(!Data=AllocMem(256*sizeof(UBYTE),MEMF_CHIP))) close
    w();
129 zv if(!SPort=CreatePort("SPort",0))) close
    w();
130 Tz Sound.ica_Request.io_Message.mn_ReplyPort =SPort;
131 uS Sound.ica_Request.io_Message.mn_Mode.ln_Pri=10;
132 O8 Sound.ica_Data =&buffer;
133 Rd Sound.ica_Length=(ULONG)sizeof(buffer);
134 FO if(SDev=OpenDevice(AUDIONAME,0,&Sound,0)) close
    w();
135 i7 Sound.ica_Request.io_Command=CMD_WRITE;
136 QU Sound.ica_Request.io_Flags =ADIOF_PERVOL IOP_QUICK;
137 BV RP =win->RPort,
138 EJO }
139 ja ULONG GetMessage() /* Mes
    sagePort auslesen */
140 Cr
141 yg4 struct IntuiMessage *mes;
142 Ko CLASS=CODE=0;
143 KE ADDR=NULL,
144 yD if(mes=GetMsg(win->UserPort)) [ CLASS=mes->Class
    ,
145 VPF CODE=mes->Code;
146 xs ADDR=mes->IAddress;
    ReplyMsg(mes);
147 NT
148 Ot4 ]
149 NO return(CLASS);
150 Qv0 }
151 Eh VOID SetPoint(x,y) /* Punkt (incl. Linie) im Zei
    chenfeld ausgeben */
152 8F4
153 Ky UBYTE y;
154 Qt0 {
155 Gw4 SHORT col;
156 hE col=2*((x==Ende)|| (x==Start));
157 6E Data[x/2]=127-y;
158 90 SetDrMd(RP,JAM1);
159 4R SetAPen(RP,1+col);
160 Pe Move(RP,x,0);
161 yb Draw(RP,x,255),

```



```

162 9k Move(RP,x+1,0);
163 Es Draw(RP,x+1,255);
164 RE SetAPen(RP, 2);
165 xG WritePixel(RP,x,y);
166 OQ WritePixel(RP,x+1,y);
167 hCO }
168 j3 VOID NewW() /* neue Werte für Period, Cycle
und Volume lesen */
169 f8 {
170 oP4 l=PLInfo.VertPot/GL,
171 3S h=PHInfo.VertPot/GH,
172 0x p=256*h+l;
173 jZ p=(p<124?124:p);
174 ZI c=64-CInfo.VertPot/GC;
175 3Q cc=(c==0?1:c);
176 RV v=64-VInfo.VertPot/GV,
177 rMO }
178 4d Play()
/* Klang abspielen */
179 pI
180 9M4 NewW();
181 xL Sound.ioa_Data =Data+Start/2;
182 d7 Sound.ioa_Length=ENDE;
183 qQ Sound.ioa_Period=p,
184 MX Sound.ioa_Cycles=c,
185 MX Sound.ioa_Volume=v,
186 u4 BeginIO(&Sound);
187 gL WaitIO(&Sound);
188 2X0
189 aS VOID Save()
190 OT {
191 Xv2 COUNT 1,j;
192 EU FILE *file;
193 MZ NewW();
194 eQ if(file=fopen(name,"w")){
195 tR5 fprintf(file,"/* -- %s - erstellt mit Gölay's SEEd
--
*/\n\n",name),
196 11 fprintf(file,"#include <devices/audio.h>\n",
197 ar fprintf(file,"#include <exec/memory.h>\n");
198 6z fprintf(file,"#define Play(); BeginIO(&Sound); WaitIO
(&Sound);\n\n");
199 tF fprintf(file,"UBYTE buffer=15,*SData,SD[%d]={\n\t",END
E);
200 Pq for(i=Start/2,j=0;i<=Ende/2;i++,j++){
201 RA9 fprintf(file,"%jd",Data[i]);
202 11 if(i!=Ende/2) fprintf(file,",");
203 30 if(j==15){ fprintf(file,"\n\t");
204 uJQ j=-1;
205 Jo9 }
206 Kp5 }
207 Rn fprintf(file,"\n);\n");
208 do fprintf(file,"extern struct MsgPort\t*CreatePort();\n",
),
209 9T fprintf(file,"extern struct Device\t*OpenDevice();\n"),
210 W2 fprintf(file,"struct MsgPort\t\t*SPort;\n");
211 H6 fprintf(file,"struct Device\t\t*SDev;\n");
212 kr fprintf(file,"struct IOAudio\t\t Sound;\n\n");
213 cM fprintf(file,"VOID SoundOff();\n\n");
214 DC fprintf(file,"tif(1SDev)\t\tCloseDevice(&Sound);\n");
215 Wo fprintf(file,"tif(SPort)\t\tDeletePort(SPort,sizeof(str
uct MsgPort));\n");
216 Os fprintf(file,"tif(SData)\t\tFreeMem(SData,%d);\n\n",
ENDE),
217 M1 fprintf(file,"SHORT SoundOn();\n\n");
218 t1 fprintf(file,"tUBYTE *AllocMem();\n");
219 8G fprintf(file,"tCOUNT i;\n\n");
220 6d fprintf(file,"tif(!SData=AllocMem(%d,MEMF_CHIP)))\tr
eturn(1);\n",ENDE);
221 RI fprintf(file,"tfor(i=0;i<%d;i++)\n\t\tSData[i]=SD[i]
,\n",ENDE),
222 y4 fprintf(file,"tif(!SPort=CreatePort(\"SPort\",0))\n\t\t
\n");
223 es fprintf(file,"t\tSoundOff();\n\t\t\treturn(1);\n\t\t\n"),
224 4h fprintf(file,"tSound.ioa_Request.io_Message.mm_ReplyP
ort=SPort;\n");
225 1K fprintf(file,"tSound.ioa_Request.io_Message.mm_Mode.1
n_Pri=10,\n",
226 fw fprintf(file,"tSound.ioa_Data=&buffer;\n");
227 kz fprintf(file,"tSound.ioa_Length=(ULONG)sizeof(buffer)
;\n");
228 zX fprintf(file,"tif(SDev=OpenDevice(AUDIONAME,0,&Sound,
0),\n"),

```

```

229 ky fprintf(file,"t\tSoundOff();\n\t\t\treturn(1);\n\t\t\n")
,
230 gC fprintf(file,"tSound.ioa_Request.io_Command=CMD_WRI
TE;\n");
231 it fprintf(file,"tSound.ioa_Request.io_Flags=ADIOF_PER
VOL+OF_QUICK;\n");
232 gf fprintf(file,"tSound.ioa_Data\t\t\t=SData,\n");
233 gn fprintf(file,"tSound.ioa_Length\t\t\t=%d;\n",ENDE);
234 8s fprintf(file,"tSound.ioa_Period\t\t\t=(UWORD)%d;\n",p),
235 3U fprintf(file,"tSound.ioa_Volume\t\t\t=%d;\n",v);
236 IS fprintf(file,"tSound.ioa_Cycles\t\t\t=(UWORD)%d;\n",c);
237 F4 fprintf(file,"treturn(0);\n\n");
238 Q7 fclose(file);
239 rM2 }
240 jn else DisplayBeep(0);
241 tOO
242 9F VOID CheckReq() /* Reques
ter kontrollieren */
243 rK {
244 Hu4 SHORT reqclear=0;
245 Uj while(reqclear==0){
246 1W8 GetMessage();
247 4j if(CLASS==GADGETUP){
248 61C switch(GAD_ID){
249 g1G case 12:{
250 JkK Save();
251 Xk reqclear=1,
252 88 break;
253 5aG
254 or case 13:{
255 boK reqclear=1;
256 3C break;
257 9eG
258 W0 default:break;
259 BgC }
260 Ch8 }
261 14
262 EJO
263 VJ VOID main()
264 Cf {
265 1f4 SHORT x=0,y=0;
266 p0 BOOL ende=0;
267 19 COUNT 1,j;
268 zg FLOAT k;
269 MO openv();
270 5t SetAPen(RP, 1);
271 V0 RectFill(RP,0,0,510,256);
272 Ib for(i=0;i<512,i+=2) SetPoint(1,127);
273 Nk while(!ende),
274 Og8 Wait(1<<win->UserPort->mp_SigBit);
275 Ba GetMessage();
276 3h switch(CLASS){
277 G7C case MOUSEBUTTONS:{
278 qOG if(CODE==MENUDOWN){
279 71K Play();
280 1q while(CODE!=MENUUP) GetMessage(),
281 K2G
282 A1 else if(CODE==SELECTDOWN){
283 FGK while(CODE==SELECTUP){
284 d80 GetMessage();
285 Om if((x!=MOUSX)! (y!=MOUSY)){
286 mtS x=MOUSX;
287 al y=MOUSY;
288 xx if(x<=510) SetPoint(2*(UBYTE){
(FLOAT)x/2.0},y),
289 f10
290 gBK }
291 hCG }
292 da break;
293 JEC }
294 7n case GADGETUP:{
295 rTH switch(GAD_ID){
296 2C /* Null */
297 hOP for(i=0;i<512,i+=2) SetPoint(1
,127),
298 ja break,
299 pKL }
300 G3 case 5:{ /* Sinus */
301 YGP for(j=0;j<2;j++){
302 1JT for(i=0,i<256,i+=2){

```

Listing 1. (Fortsetzung)

```

303 DYX      k=sqrt((FLOAT) ((256-1)*(
304 ob        256-1)+1*1));
305 nu        k=j?-k+128.0:k-128.0;
              SetPoint(1+256*j,(UBYTE)k)
              ;
              }
              break;
          ]
          case 6: { /* 3Eck */
              for(j=0;j<2;j++){
                  for(i=j*256;i<128+j*256;i+=2)
                  {
                      k=(FLOAT)i*.7;
                      k=j?k+204.0:k-128.0;
                      SetPoint(1,(UBYTE)k);
                  }
                  for(i=128+j*256;i<256+j*256;i
                      +=2)
                      k=(FLOAT)i*.7;
                      k=j?-k-28:k-52.0;
                      SetPoint(1,(UBYTE)k);
                  }
              }
              break;
          }
          case 7: { /* SZahn */
              for(j=0;j<2;j++){
                  for(i=j*256;i<128+j*256;i+=2)
                      k=.7*(FLOAT)(j*256+180-1);
                      SetPoint(1,(UBYTE)k);
                  }
                  for(i=128+j*256;i<256+j*256;i
                      +=2)
                      k=.7*(FLOAT)(j*256+436-1);
                      SetPoint(1,(UBYTE)k);
                  }
              }
              break;
          }
          case 8: { /* Recheck */
              for(j=0;j<2;j++){
                  for(i=j*256;i<128+j*256;i+=2)
                      SetPoint(1,38);
                  for(i=128+j*256;i<256+j*256;i
                      + 2) SetPoint(1,218);
                  }
              }
              break;
          }
          case 9: { /* 2mal */
              for(i=j=0;i<256;i+=2,j++) Data[j]
              =Data[i];
              for(i=128;i<256;i++) Data[i]
              =Data[i-128];
              for(i=0,i<512;i+=2) SetPoint(1,(B
              YTE)(127-Data[i/2]));
              break;
          }
          case 10: { /* Save */
              Request(&Req,win);
              CheckReq();
              break;
          }
          case 11: {
              ende=1;
              break;
          }
          default: break;
      }
      break;
  }
  case VANILLAKEY:
      switch(CODE){
          case 49: { /* Ende links */
              if((Ende>2)&&((Ende-2)>Start)){
                  Ende-=2;
                  SetPoint(Ende+2,(BYTE)(127-Dat
                  a[(Ende+2)/2]));
                  SetPoint(Ende,(BYTE)(127 Data[
                  Ende/2]));
              }
          }
          case 51: { /* Ende rechts */
              if(Ende<510){
                  Ende+=2;
                  SetPoint(Ende-2,(BYTE)(127-Dat
                  a[(Ende-2)/2]));
                  SetPoint(Ende,(BYTE)(127-Data[
                  Ende/2]));
              }
              break;
          }
          case 52: { /* Ende links schnell */
              if((Ende>8)&&((Ende-8)>Start)){
                  Ende-=8;
                  SetPoint(Ende+8,(BYTE)(127-Dat
                  a[(Ende+8)/2]));
                  SetPoint(Ende,(BYTE)(127-Data[
                  Ende/2]));
              }
              break;
          }
          case 54: { /* Ende rechts schnell */
              if(Ende<506)
                  Ende+=8;
                  SetPoint(Ende-8,(BYTE)(127-Dat
                  a[(Ende-8)/2]));
                  SetPoint(Ende,(BYTE)(127 Data[
                  Ende/2]));
              }
              break;
          }
          case 50:
          case 53: { /* Ende-Position */
              i=Ende;
              Ende=510;
              SetPoint(i,(BYTE)(127-Data[i/2]));
              SetPoint(510,(BYTE)(127-Data[255]
              ));
              break;
          }
          case 55: { /* Start links */
              if(Start>=8)
                  Start-=8;
                  SetPoint(Start+8,(BYTE)(127-Da
                  ta[(Start+8)/2]));
                  SetPoint(Start,(BYTE)(127-Data
                  [Start/2]));
              }
              break;
          }
          case 57: { /* Start rechts */
              if((Start<504)&&((Start+8)<Ende)
              ){
                  Start+=8;
                  SetPoint(Start-8,(BYTE)(127-Da
                  ta[(Start-8)/2]));
                  SetPoint(Start,(BYTE)(127-Data
                  [Start/2]));
              }
              break;
          }
          case 56: { /* Start-Position */
              i=Start;
              Start=0;
              SetPoint(i,(BYTE)(127-Data[i/2]));
              SetPoint(0,(BYTE)(127-Data[0]));
              break;
          }
          default: break;
      }
      break;
  }
  default: break;
}
close();
}
(C) 1989 M&T

```

Listing 1. (Schluß)


```
#include <devices/audio.h> /* notwendige Headerfiles einbinden */
#include <exec/memory.h>
#define Play, BeginIO(&Sound), WaitIO(&Sound), /*
    Abspielen des Sounds */

UBYTE buffer=15,*SData,SD[56]=
    /* globale Variablen und Sounddaten hier 56 */
    1, 7, 13, 18, 24, 29, 35, 41, 46, 52, 57, 63, 69, 74, 80, 85,
    168,173,179,185,190,196,201,207,213,218,224,229,235,241,246,252,
    1, 7, 13, 18, 24, 29, 35, 41, 46, 52, 57, 63, 69, 74, 80, 85,
    168,173,179,185,190,196,201,207

extern struct MsgPort *CreatePort,
extern struct Device *OpenDevice(),
struct MsgPort *SPort,
struct Device *SDev,
struct IOAudio Sound,

VOID SoundOff() /* Ressourcen freigeben */
{
    if(SDev, CloseDevice &Sound), /* Sounddevice schließen */
    if(SPort) DeletePort(SPort,sizeof struct MsgPort, /* Port */
    /* Port */
    if(SData, FreeMem SData,56), /* Speicherbereich
    freigeben */
}

SHORT SoundOn() /* Klang initialisieren, Rückgabewert in
    Fehlerfälle 1 *

    UBYTE *AllocMem();
    COUNT i,

    /* Mit AllocMem() Speicher für Daten im Chipmemory anfordern */

    if( SData=AllocMem(56,MEMP_CHIP), return 1,
```

```
for(i=0;i<56,i++)
    SData[i]=SD[i], /* Daten ins Chipmemory kopieren */
if(!SPort=CreatePort("SPort",0)) /* Antwortport kreieren */

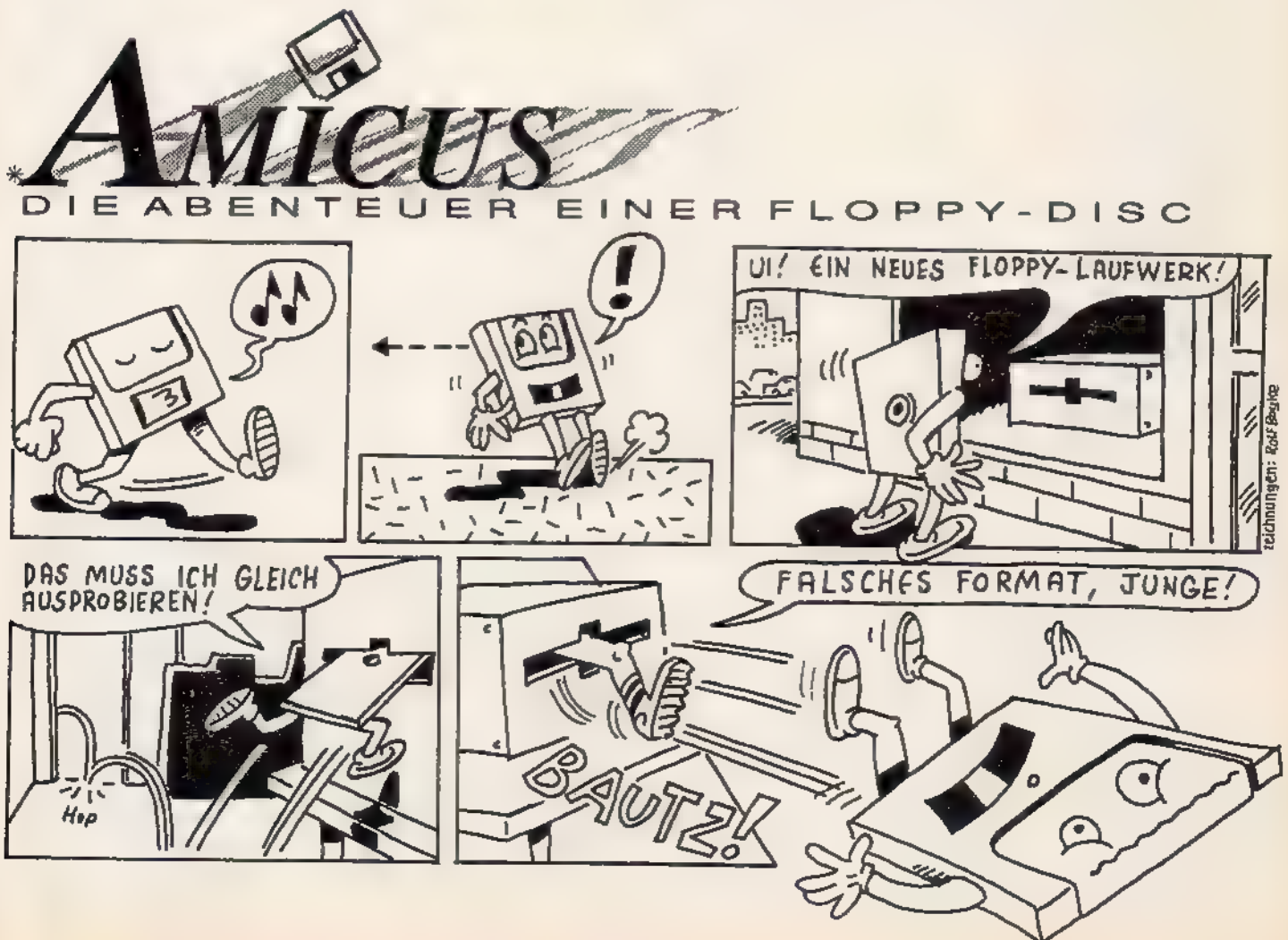
    SoundOff,
    return 1),

Sound.ioa_Request.io_Message mn_ReplyPort=SPort,
/* Portzuweisung */
Sound.ioa_Request.io_Message mn_Node,ln_Pri=10,
/* Prozesspriorität */
Sound.ioa_Data=&buffer, /* Soundkanäle festlegen */
Sound.ioa_Length= 56*5, sizeof(buffer),
/* Größe des Buffers */
/* SDev=OpenDevice(AUDIONAME,0,&Sound,0)

/* Audio-Device öffnen */
SoundOff(), /* hat nicht geklappt */
return 1, /* daher zurück mit 1 */

Sound.ioa_Request.io_Command =CMD_WRITE,
/* Ausgabe vorbereiten */
Sound.ioa_Request.io_Flags =ADIOF_PEP,OLIOF_CLICK,
/* Flags */
Sound.ioa_Data =SData,
/* Zeiger auf die Sounddaten */
Sound.ioa_Length =56,
/* Anzahl der Werte (UBYTE) */
Sound.ioa_Period =*(UWORD)255, /* Samplerperiode */
Sound.ioa_Volume =64, /* Lautstärke */
Sound.ioa_Bytes =*UWORD,1, /* Wiederholungen */
return 0, /* Null: alles OK */
```

Listing 2. Ein dokumentiertes Listing zum Quellcode einer Sounddatei, die vom SEEd erzeugt wurde



ETIKETTEN

Meist steht auf den Diskettenaufklebern nur ein Name. Werden es aber mehr und mehr Disketten, verliert man den Überblick. Die Zeit, die man für das Suchen verschwendet, wird immer länger. Etiketten für 3-1/2-Zoll-Disketten mit den Namen der wichtigsten Dateien helfen das Gesuchte schneller zu finden.

Um diese Etiketten zu drucken, gibt es verschiedene Wege. Einen wirklich komfortablen geht »DiskEti«. Das Einlesen der Dateinamen geschieht durch das Programm. Somit entfällt schon viel Tipparbeit. Welche Namen auf das Etikett gedruckt werden, entscheiden Sie durch Anklicken mit der linken Maustaste. Die Namen können auf Wunsch editiert werden. All diese Gründe sprechen für die Wahl von DiskEti zum Programm des Monats.

Für manche Routinen braucht Amiga-Basic die Unterstützung des Betriebssystems. Damit das funktioniert, benötigt DiskEti die vier Dateien:

- ☐ dos.bmap ☐ exec.bmap
- ☐ graphics.bmap ☐ intuition.bmap

Sie finden diese Dateien auf Ihrer Extras-Diskette. Nicht vorhandene »bmap«-Dateien generieren Sie wie im Artikel »Neue Dimensionen in Basic«, Ausgabe 1/89, Seite 147, beschrieben. Kopieren Sie die Dateien in dasselbe Verzeichnis, in dem DiskEti und Amiga-Basic stehen. Nach dem Start erscheint zunächst eine Kurzanleitung zum Programm. Nach einem Klick mit der linken Maustaste sehen Sie die Arbeitsfläche des Programms (Bild 1). In dem Fenster sind alle benötigten Schalter zu sehen. Es werden keine Pull-Down-Menüs verwendet.

Es ist gut, viele Disketten mit vielen Programmen zu haben. »DiskEti« erleichtert Ihnen durch Etiketten die Suche in Ihren Beständen. Komfortable Bedienung ist dabei Trumpf.



den Namen der Datei auf Diskette, sondern nur den Namen für den Ausdruck.

Mit dem Schalter »nächste Datei« übergehen Sie Dateien, die nicht auf den Aufkleber gedruckt werden sollen. Genauso ist »nächstes Dir« zu verwenden. Es läßt ein gesamtes Unterverzeichnis aus. Das ist sinnvoll, wenn es sich zum Beispiel um ständig wiederkehrende Verzeichnisse wie LIBS: handelt.

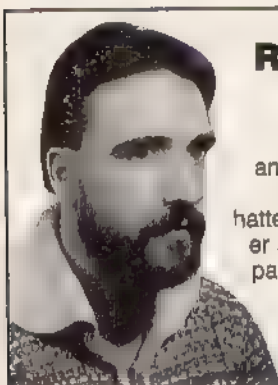
Nachdem Sie nun den Diskettennamen festgelegt haben, folgen die einzelnen Dateien. Dabei werden Unterverzeichnisse, »info«-Dateien und »fastdir«-Dateien nicht berücksichtigt. Als erstes zeigt Ihnen DiskEti die Dateien im Hauptverzeichnis der Reihe nach an. Sind sie bearbeitet, folgen die Dateien in den Unterverzeichnissen. Pro Subdirectory kann das Programm 100 Einträge verwalten, die vor der Anzeige alphabetisch sortiert werden.

Sollten Sie beim Arbeiten merken, daß sich die falsche Diskette im Laufwerk befindet, ist das kein Problem. Durch Aktivieren von »andere Diskette« kehrt DiskEti zum Programmstart zurück. Dann legen Sie die richtige Diskette ein und beginnen von vorne.

Wenn Sie den Ausdruck starten wollen, betätigen Sie den Schalter »ETIKETT DRUCKEN«. Es erscheint ein Requester, mit dem Sie entweder den Ausdruck beginnen oder zum Programmstart zurückgehen. Ist das Etikett voll (32 Einträge), bringt DiskEti automatisch den Requester. Dies geschieht auch, wenn auf der Diskette kein Eintrag mehr zu finden ist.

Beendet wird das Programm durch Betätigung des Schließsymbols (close gadget) links oben.

DiskEti arbeitet in der abgedruckten Version mit Epson-kompatiblen Druckern. Das Umstellen auf andere Drucker ist, dank der sauberen Programmierung, aber kein Problem. Die für Ihren



RÜDIGER HOFNER

Rüdiger Hofner, der Autor von »DiskEti«, bezeichnet sich selbst als »Opa am Computer« und das im Alter von 30 Jahren. Den ersten Computerkontakt hatte er Ende 1984 mit einem VC 20, den er sich kaufte, um seinen Kindern in ein paar Jahren nicht unwissend in Sachen Computer gegenüberstehen zu müssen. Aus dem »nur mal sehen« wurde ein richtiges Hobby. Als nächster Computer folgte, wie sollte es anders sein, ein C 64. Im Jahr

1987 war es dann soweit, ein Amiga kam ins Haus. Sein nächster Schritt wird das Lernen der Programmiersprache C sein. Von dem 2000-Mark-Gewinn (Programm des Monats im AMIGA-Magazin 2/89) kaufte er eine Festplatte und gute Software.

Das Programm wartet nun auf die Auswahl eines Laufwerks. DiskEti unterstützt die Laufwerke DF0: und DF1:. Klicken Sie einen der beiden Schalter (»df0:« oder »df1:«) an. Im oberen Textfeld am rechten Fensterrand sehen Sie nun den Namen der Diskette. Wie später auch die einzelnen Programmnamen, kann er editiert werden. Um das zu tun, klicken Sie den Schalter »Name ändern« an. Der Name erscheint im Textfeld darunter. Dort kann er verändert werden. Folgende Tastenkombinationen stehen Ihnen dabei zur Verfügung:

- < Shift - Cursor links > : Der Cursor springt an den Anfang.
- < Shift - Cursor rechts > : Der Cursor wird an das Ende gesetzt.
- < rechte Amiga - x > : löscht die Zeichenkette.

Die Tasten < Del > und < Backspace > sind wie gewohnt belegt. Mit < RETURN > übernehmen Sie den Namen. Wollen Sie ihn verbessern, klicken Sie erneut auf »Name ändern«.

Sieht die Zeichenkette wie gewünscht aus, klicken Sie den Schalter »übernehmen« an. Der Name wird dann in das unten dargestellte Etikett eingetragen. Natürlich ändern Sie damit nicht

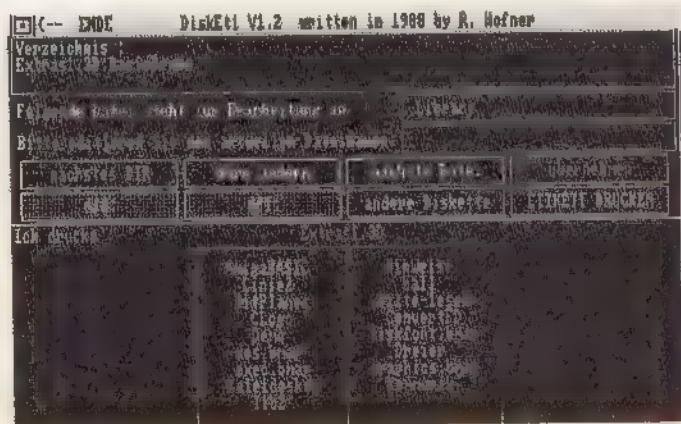


Bild 1. Das ist »DiskEti« bei der Auswahl der Dateinamen

ANWENDUNGEN

MIT PFIFF

Drucker gültigen Sequenzen finden Sie im Handbuch. Folgende Zeichenketten werden verwendet, um den Drucker anzusteuern:

- esc\$: das Escape-Zeichen, das Steuersequenzen einleitet;
- init\$: Steuersequenz zum Initialisieren des Druckers;
- fett\$: Einschalten des Fettdrucks;
- abstand\$: dient zur Festlegung des Zeilenabstands in $\frac{1}{2}$ Zoll. Hier ist bei einigen Druckern eine zweite Sequenz nötig, um den eingestellten Abstand zu aktivieren. Beim Star NL-10 funktioniert dies so:

```
PRINT #1,esc$;"A";CHR$(6)
```

```
PRINT #1,esc$;"2"
```

— klein\$: Stellt die Schriftart Superscript (hochgestellt) ein. In Verbindung mit dem geänderten Zeilenabstand passen so mehr Daten auf das Etikett.

— eng\$: Druck in Schmalschrift (132 Zeichen pro Zeile);

— elite\$: die Sequenz, die den Drucker anweist, in der Schriftart Elite zu drucken (12 Zeichen pro Zoll);

— Ende\$: stellt wieder den normalen Zeilenabstand her.

Die Festlegung dieser Zeichenketten geschieht in dem Unterprogramm »Etiket« ab Zeile 627. Die entsprechenden Sequenzen entnehmen Sie Ihrem Druckerhandbuch. Doch dies ist wie ge-

sagt nur notwendig, wenn Ihr Drucker mit der vorliegenden Version nicht richtig arbeitet.

Die Endlosetiketten für $3\frac{1}{2}$ -Zoll-Disketten sind in jedem gut sortierten Schreibwarenhandel erhältlich. R. Hofner/R. Beaupoll/kn

LabelMaker

AmigaBasic	Etikett-Ed
AddBuffers	Assign
BindDrivers	Break
CD	ChangeTaskPri
Copy	Date
Delete	Dir
DiskChange	DiskDoctor
Echo	Ed
Edit	Else
EndCLI	EndIf
Execute	FailAt
Fault	FileNote
If	Info
Install	Join
Lab	List
LoadWB	MakeDir

Bild 2.
Ein Aus-
druck von
»DiskEti«
mit einem
Epson-
Drucker

Programmname: DiskEti

Computer: A500, A1000, A2000
mit Kickstart 1.2 & 1.3

Sprache: Amiga-Basic 1.2

Programmautor: Rüdiger Hofner

```

1 :MO CLEAR,30000
2 eG DEFING a-z
3 tv REM ###
4 iU SCREEN 1,640,256,2,2
5 kJ PALETTE 0,0,0,1
6 iD PALETTE 1,9,9,9
7 qv PALETTE 2,0,0,0
8 JW LIBRARY "intuition.library"
9 J9 LIBRARY "graphics.library"
10 Hw WINDOW CLOSE 1
11 J5 WINDOW 2,,0,1
12 nr Fenster&=WINDOW 7
13 ey rOREW Fenster&+98,0
14 B1 CALL RefreshWindowFrame&Fenster&
15 Hm Meldung
16 op1 js$=CHR$ 252,
17 JK LOCATE 3,18
18 RJ Schatten "Herzlich willkommen zu DiskEti",0
19 FI PRINT
20 D1 Schatten "Vorweg erst mal ein paar kleine Anmerkungen:",0
21 Uy Schatten "Ich lese die Verzeichnisse einer",1
22 4p Schatten " von Ihnen bestimmten Diskette",0
23 6V Schatten "Dabei ist folgendes zu beachten:",0
24 kN PRINT
25 GS Schatten "- Ich lese maximal 100 Files pro Verzeichnis",0
26 M2 Schatten "- Ich zeige keine Directory-Namen in der Auflis-
ung an",0
27 e2 Schatten "- Ich zeige keine .info-Dateien in der Auflis-
g an",0
28 rQ Schatten "- Die Namen sind maximal 15 Zeichen lang",0
29 pS PRINT
30 TM Schatten "Die Dateien werden alphabetisch sortiert und ein-
geordnet",1
31 i4 Schatten " angezeigt. Dann können ",0
32 YR Schatten "Sie diesen Dateinamen editieren, "+ue$+"bersprin-
gen",1
33 wH Schatten " zum Druck "+ue$+"bernehmen",0
34 7s Schatten "oder aber direkt in das nächste Unterverzeichnis
" 1
35 ep Schatten "wechseln. Das Wechseln ",0

```

```

36 7T Schatten "passiert automatisch am Ende eines Verzeichnisse
s. ",1
37 7s Schatten "Von Ihnen "+ue$+"bernommene",0
38 QM Schatten "Dateien werden ständig angezeigt!",0
39 i4 Schatten "Der Name der Diskette wird auch zum Druck "+ue$+
"bernommen!",0
40 i4 Schatten "(Den können Sie aber auch ändern.)",0
41 i4 PRINT
42 oM Schatten "ACHTUNG!!!! Ich "+ue$+"bernehme maximal 32 D
ateinamen",0
43 J2 PRINT
44 i4 Schatten ".....linke Maustaste dr"+ue$+"oken.....",0
45 Va Maus
46 iO CLS
47 60 LOCATE 12,29
48 xM Schatten "Einen Moment bitte !",0
49 DV LOCATE 14,18
50 ex Schatten "Ich muß erst mal n paar .bmaps lesen. ....",0
51 ONO LIBRARY "dos.library"
52 qT LIBRARY "exec.library"
53 iJ DECLARE FUNCTION ExNext% LIBRARY
54 6Q DECLARE FUNCTION Look LIBRARY
55 9d DECLARE FUNCTION AllocMem LIBRARY
56 e1 DECLARE FUNCTION IoErr% LIBRARY
57 01 DECLARE FUNCTION GetMag LIBRARY
58 kJ DECLARE FUNCTION OpenWindow LIBRARY
59 iR DECLARE FUNCTION Examine% LIBRARY
60 26 DECLARE FUNCTION ViewAddress LIBRARY
61 rb DECLARE FUNCTION ViewPortAddress LIBRARY
62 AJ DECLARE FUNCTION AutoRequest LIBRARY
63 TC va.iab.en
64 qd1 zaehler% = 1
65 b1 SpeicherListe = 0&
66 5m i% = 0
67 yM x% = 100
68 iQ y% = 100
69 Ea .o1% = 3
70 m4 DIM SHARED dir.Name$(x%)
71 ay DIM SHARED dir.Type$(x%)
72 F2 DIM SHARED info$(x%), Druck$(x%)
73 3x DIM a$(y%)
74 84 DIM a$(y%)

```

Listing. Mit »DiskEti« bedrucken Sie Etikettenaufkleber schnell und komfortabel. Bitte mit dem Checksummer (Seite 143) eingeben.

```

75 VGO Fenster:
76 e1f MacheRand FeldRand,-1,-1,1,150,14
77 VW MacheRand AusgRand,-1, 1,3,630,28
78 O- MacheRand AusgRand1, 1,-1,3,260,14
79 NL IntText EditTxt,1,1,31,2,"Name ändern",0%
80 VQ IntText WeiterTxt,1,1,23,2,"nächste Datei",0%
81 oV IntText OkTxt,3,1,35,2,"Übernehmen",0%
82 tO IntText GoonTxt,1,1,27,2,"nächstes DIR",0%
83 + IntText dfOTxt,1,1,59,2,"df0:",0%
84 z2 IntText df1Txt,1,1,59,2,"df1:",0%
85 xF IntText DiskTxt,1,1,15,2,"andere Diskette",0%
86 Kh IntText DruckTxt,1,1,15,2,"ETIKETT DRUCKEN",0%
87 w2 IntText FrageTxt,1,2,10,8,"Wollen Sie wirklich aufordern ??",0%
88 Z7 IntText JaTxt,1,2,6,3,"Ja !",0%
89 ET IntText NeinTxt,1,2,6,3,"Nein",0%
90 rK IntText DruckFrageTxt,1,2,10,8,SPACES(11)+"Drucken ??",0%
91 D- FeldDef GoonFeld,0%,12,70,148,12,0,1,1,FeldRand,GoonTxt,0%
92 Gc FeldDef EditFeld,GoonFeld,167,70,148,12,0,1,1,FeldRand,Edit
93 an FeldDef WeiterFeld,EditFeld,322,70,148,12,0,1,1,FeldRand,Weiter
94 ya FeldDef OkFeld,WeiterFeld,477,70,148,12,0,1,1,FeldRand,OkT
95 fr FeldDef dfOFeld,OkFeld,12,85,148,12,0,1,1,FeldRand,dfOTxt,
96 dj FeldDef df1Feld,dfOFeld,167,85,148,12,0,1,1,FeldRand,df1Tx
97 sA FeldDef DiskFeld,df1Feld,322,85,148,12,0,1,1,FeldRand,Disk
98 ix FeldDef DruckFeld,DiskFeld,477,85,148,12,0,1,1,FeldRand,Dr
99 xC titel$="<-- ENDE"
100 P8 titel$=titel$+SPACES(8)+"DiskEti V1.2 "
101 w4 titel$=titel$+" written in 1988 by R. Hofner"
102 jw WINDOW CLOSE 2
103 lo WINDOW 3,"",0,99)-(629,242),0,1
104 WYO sM # # #
105 Co2 FensterDef N Fenster,0,0,639,100,64+512%,84+4096%,DruckFeld
106 k- FensterAdresse=ipow(window,Fenster)
107 qP IF FensterAdresse=0 THEN ERROR 7
108 qP FensterRP=PEEK(L(FensterAdresse+50))
109 k5 FensterUP=PEEK(L(FensterAdresse+56))
110 lo DrawBorder FensterRP,AusgRand,5,11
111 yC DrawBorder FensterRP,AusgRand1,375,40
112 le DrawBorder FensterRP,AusgRand1,375,55
113 yU CALL SetAPen(FensterRP,1) 'Schriftfarbe f. Druckl auf norm
114 qC druckl "Folgende Datei steht zur Bearbeitung an ",-1
115 xD druckl "Bitte editieren Sie den aktuellen Dateinamen ",-
116 to GOSUB ZeichneEtikett
117 aD0 Hauptprogramm.
118 Jx3 GOSUB FelderEin
119 m- Fuehle 6,11,631,36
120 fa druckl "Bitte wählen Sie das Laufwerk aus .. ",9,19
121 Kh GOSUB Meldungsschleife
122 xp Fuehle 6,11,631,36
123 ZVO LiesDir:
124 Fd3 FOR schleife% = 1 TO zaeher%-1
125 2A4 IF ak(schleife%) = stufe% THEN
126 BZ6 such$=a$(schleife%)
127 e7 z% = x%
128 l3 zeige%=0
129 w4 Fuehle 6,11,631,36
130 n5 druckl "Ich lese : ",9,11:druckl a$(such$),9,19
131 Jo druckl "Bitte ein wenig Geduld.....",9,27
132 eR LiesDisk such$, x%
133 A1 IF fehler%=500 THEN
134 CJ7 neu%=1
135 fo GOTO nochmal
136 816 neu%=0
137 w4 max% = x%
138 n1 x% = z%
139 wY sort
140 j- IF schleife%=0 THEN
141 Kk8 a$ = dir$(Name$) + ". "
142 lQ druckl "IF-18 e2",Lr a$
143 IA Fuehle 6,11,631,36
144 TI druckl "Diskette name ",9,19
145 GQ IF LEN(Druck$(0)) > 15 THEN

```

```

146 K1A Druck$(0)=LEFT$(Druck$(0),15)
147 JCS END IF
148 Sa druckl Druck$(0),388,41
149 8o eds=Druck$(0)
150 kA GOSUB Meldungsschleife
151 Gz IF neu% = 1 THEN GOTO nochmal
152 DH6 END IF
153 Jn z% = a$(schleife%)
154 lC dir$ = LEFT$(z$,LEN(z$)-1)
155 UM Fuehle 6,11,631,36
156 Pz druckl "Verzeichnis : ",9,11:druckl dir$,9,19
157 cz FOR anz% = 1 TO max%
158 bHB IF dir$.Typ$(anz%) <> "DIR" THEN
159 yJD IF RIGHT$(dir$.Name$(anz%),5) <> ".info" THEN
160 Sx7 IF RIGHT$(dir$.Name$(anz%),8) <> ".fastdir"
161 xOH info$(zeige%)=dir$.Name$(anz%)
162 Si zeige%=zeige%+1
163 ZSF END IF
164 sTO END IF
165 GzB ELSE
166 rxD Fl=1
167 f- e2 "schleife% a$ schleife% dir Name$(anz%) + ",
168 At ak(zaeher%)=stufe%+1
169 oo zaeher%=zaeher%+1
170 GZB END IF
171 vq6 NEXT anz%
172 l64 END 1
173 JS ausgabe$=
174 nK Abfrage:
175 WPS IF ausgabe% < zeige% THEN
176 e47 info$(ausgabe%)=LEFT$(info$(ausgabe%),15)
177 Op Fuehle 380,40,631,50
178 xL druckl info$(ausgabe%),388,41
179 Gc eds=info$(ausgabe%)
180 8e GOSUB Meldungsschleife
181 Wp IF neu% = 1 OR drucken% = 1 THEN GOTO nochmal
182 Sa ausgabe%=ausgabe%+1
183 w2 GOTO Abfrage
184 um3 END IF
185 2yl NEXT schleife%
186 ha e2 " = a schleife%
187 QG IF neu% = 1 THEN
188 8D- =
189 JW stufe% =stufe%+1
190 e9 GOTO LiesDir
191 lu1 END 1
192 TyO nochmal:
193 jpi IF neu% = 0 THEN
194 bC3 Etikett
195 3y1 END IF
196 J8 Fuehle 380,40,631,50
197 la Fuehle 9,84,317,97
198 Ct GOSUB neuinit
199 Ug GOTO Hauptprogramm
200 6VO ALesAus:
201 ra1 CALL CloseWindow(FensterAdresse)
202 GE CALL SpeicherFrei
203 YF WINDOW CLOSE 3
204 EAD WEF # # #
205 5A1 SCREEN CLOSE 1
206 yC LIBRARY CLOSE
207 5D END
208 cyO Meldungsschleife:
209 Jx1 Lauf%=
210 Yg Hauptschleife:
211 rF IF Lauf% = 1 THEN
212 Oe3 IntuiMeldung = GetMsg(FensterUP)
213 we IF IntuiMeldung > 0 THEN GOSUB IntuitionMeldung
214 Ct GOTO Hauptschleife
215 P13 END IF
216 qSO RETURN
217 Ab IntuitionMeldung.
218 ael MeldungsTyp=PEEK(L(IntuiMeldung+20))
219 ei Platz=PEEK(L(IntuiMeldung+28))
220 76 FeldNr=PEEK(Platz+39)
221 A- MeldungsTyp=64% THEN
222 J- IF FeldNr% = 1 THEN
223 8- a$ =
224 8- a$ =
225 8- a$ =
226 8- a$ =
227 8- a$ =

```



```

228 ZE      FeldDef StrGad,0%,389,57,150,12,0,1,4,0%,0%,EString,9
229 OG      CALL AddGadget(FensterAdresse,StrGad,-1)
230 u5      CALL RefreshGadgets(StrGad,FensterAdresse,0%)
231 LX      CALL ActivateGadget(StrGad,FensterAdresse,0%)
232 GQ3     ELSEIF FeldNr% = 3 THEN
233 ej6      Lauf%=0
234 MX3     ELSEIF FeldNr% = 4 THEN
235 em6      IF Druck% > 0 THEN
236 Hf8      Druck$(Druck%)=info$(ausgabe%,
237 le6      END IF
238 WT      GOSUB PrintOut
239 VR      Druck%=Druck%+1
240 lq      Lauf%=0
241 ms      IF Druck% = 33 THEN drucken% = 1
242 YK3     ELSEIF FeldNr% = 5 THEN
243 o16      Lauf%=0
244 SW      a$ 0 = "if0 "
245 Yy      GOSUB FelderAus
246 gt3     ELSEIF FeldNr% = 6 THEN
247 sK6      Lauf%=0
248 ce      a$ 1 = "d1 "
249 e2      GOSUB zeGerAus
250 o23     ELSEIF FeldNr% = 7 THEN
251 p06      Lauf% = 1
252 wb      Lauf% = 0
253 vA3     ELSEIF FeldNr% = 8 THEN
254 ze6      Lauf%=0
255 NG      drucken% = 1
256 213     ELSEIF FeldNr% = 9 THEN
257 T26      LiesEdit EString,r$
258 T8      IF Druck% < 1 THEN
259 NQ8      Druck$(0,r$
260 LA      Fuehle 380,40,631,50
261 HJ      druck1 Druck$(0,,388,41
262 yU      CALL RemoveGadget,FensterAdresse,StrGad)
263 Eg      CALL RefreshGadgets(GoonFeld,FensterAdresse,0%,
264 ra6      ELSE
265 Jm8      info$(ausgabe%)=r$
266 RG      Fuehle 380,40,631,50
267 Dy      druck1 info$(ausgabe%),388,41
268 a2      CALL RemoveGadget,FensterAdresse,StrGad)
269 Km      CALL RefreshGadgets(GoonFeld,FensterAdresse,0%)
270 JB6      END IF
271 JC      Fuehle 376,55,631,66
272 KD3      END IF
273 L21      END IF
274 K6      IF MeldungsTyp=512% THEN
275 D13      Antwort=AutoRequest(FensterAdresse,FrageTxt,JaTxt,NeinTx
t,0%,0%,300,50,
276 UJ      IF Antwort > 0 THEN GOTO AllesAus
277 P11      END IF
278 qS0      RETURN
279 lJ      PrintOut,
280 nS1      IF Druck% = 0 THEN
281 l23      lo%=0
282 PJ      COLOR 3,0
283 nJ      LOCATE 1,15
284 v8      v=INT((50-LEN(Druck$(Druck%)))/2)
285 RM      PRINT SPC(v);Druck$(Druck%),
286 N2      COLOR 1,0
287 Ex1      ELSE
288 KB3      IF lo%=1 THEN
289 u09      LOCATE lo1%,2
290 H03      ELSE
291 y49      LOCATE lo1%,24
292 ex3      END IF
293 DU      v=INT((15-LEN(Druck$(Druck%)))/2)
294 Yk      n=INT((15-LEN(Druck$(Druck%)))/2+.5)
295 ee      PRINT SPC(v);Druck$(Druck%);SPC(n);CHR$(32);
296 ..      lo%=lo%+1
297 XP      IF lo%=2 THEN
298 lJ4      lo%=0
299 nC      lo1%=lo1%+1
300 mf3      END IF
301 ng1      END "
302 E20      RETURN
303 ..      neu="i"
304 oA1      ne.% = 0
305 Ke      Druck% = 0
306 ea      stufe% = 0
307 AL      zaehler% = 1
308 d8      jetzt% = 0
309 YK      i% = 0

```

```

310 wW      senleife% = 0
311 xa      lol% = 3
312 dq      drucken% = 0
313 4z      GOSUB ZeichneEtikett
314 Q20      RETURN
315 fw      FelderEin
316 OM1      CALL OnGadget(df0Feld,FensterAdresse,0%)
317 SR      CALL OnGadget(df1Feld,FensterAdresse,0%)
318 K4      CALL OffGadget(EditFeld,FensterAdresse,0%,
319 ET      CALL OffGadget(WeiterFeld,FensterAdresse,0%)
320 9g      CALL OffGadget(OkFeld,FensterAdresse,0%)
321 sp      CALL OffGadget(GoonFeld,FensterAdresse,0%)
322 WL      CALL OffGadget(DiskFeld,FensterAdresse,0%,
323 4K      CALL OffGadget(DruckFeld,FensterAdresse,0%)
324 Ra      CALL RefreshGadgets(df1Feld,FensterAdresse,0%)
325 bD0      RETURN
326 pJ      FelderAus
327 B11      CALL OffGadget(df0Feld,FensterAdresse,0%)
328 Dr      CALL OffGadget(df1Feld,FensterAdresse,0%)
329 7u      CALL OnGadget(EditFeld,FensterAdresse,0%)
330 ko      CALL OnGadget(WeiterFeld,FensterAdresse,0%)
331 JJ      CALL OnGadget(OkFeld,FensterAdresse,0%,
332 e8      CALL OnGadget(GoonFeld,FensterAdresse,0%)
333 96      CALL OnGadget(DiskFeld,FensterAdresse,0%)
334 4e      CALL OnGadget(DruckFeld,FensterAdresse,0%)
335 S9      Fuehle 4,69,632,97
336 2K      CALL RefreshGadgets(DruckFeld,FensterAdresse,0%)
337 nP0      RETURN
338 Iv      ZeichneEtikett
339 171      CLS
340 Yw      LINE (171,10)-(461,151),3,b
341 5D      LINE (172,10)-(172,151),3
342 aw      LINE (314,10)-(314,151),3
343 28      LINE (315,10)-(315,151),3
344 y6      LINE (460,10)-(460,151),3
345 ae      PRINT "Ich drucke : "
346 wY0      RETURN
347 2Q      Ende:
348 g7      *****
****
349 29      '**          Jetzt kommen die Sub s !
**
350 19      *****
****
351 w4      REM Speicherbereiche reservieren:
352 X4      '...
353 1x      SUB ReservSpeicher(Puffer, Groesse) STATIC
354 791      SHARED SpeicherListe
355 kA      Groesse=Groesse+8
356 4U      Puffer=AllocMem(Groesse,65538%,
357 nD      IF Puffer>0 THEN
358 173      POKEL Puffer,SpeicherListe
359 To      POKEL Puffer++,Groesse
360 2L      SpeicherListe=Puffer
361 11      Puffer=Puffer+8
362 RA1      ELSE
363 RY3      ERROR 7
364 oh1      END IF
365 vx0      END SUB
366 TO      REM Speicherbereiche wieder freigeben
367 y1      -----
368 A0      SUB SpeicherFrei STATIC
369 M01      SHARED SpeicherListe
370 vU      freigeben
371 f13      IF SpeicherListe>0 THEN
372 rJ5      Adresse=PEEKL(SpeicherListe)
373 Af5      Groesse=PEEKL(SpeicherListe+4)
374 mW      FreeMem SpeicherListe, Groesse
375 NH      SpeicherListe = Adresse
376 r2      GOTO freigeben
377 1u3      END IF
378 8A0      END SUB
379 1K      REM NewWindow neu auflegen
380 Wd      '-----

```

Listing.
»DiskEti«
(Fortsetzung)



```

381 mK SUB FensterDef(bs, x%, y%, b%, h%, IDCMP, f, Feld, T%) STAT
      IC
382 b% Groesse = 48+LEN(T%)+1 'benötigte Speichergröße f. Fensterstruktur
383 LY ReservSpeicher bs,Groesse
384 ex POKEW bs,x% 'xKoordinate linke obere Ecke
385 y% POKEW bs+2,y% 'yKoordinate linke obere Ecke
386 b% POKEW bs+4,b% 'Breite
387 h% POKEW bs+6,h% 'Höhe
388 v% POKEW bs+8,65535% 'Fein- und Grobstift
389 f% POKEW bs+10,IDCMP 'Nachrichten-Flags
390 m% POKEW bs+14,f 'Fenster-Flags
391 w% POKEW bs+18,Feld 'erstes Gadget
392 t% POKEW bs+26,bs+48 'Titel
393 BDO REM ###
394 XZ WINDOW 3
395 pt POKEW bs+30,PEEK(WINDOW(7)+46) 'Zeiger auf Basic-Screen
396 vn1 POKEW bs+46,15 'Screen-Typ
397 fo FOR i%=1 TO LEN(T%)
398 f% POKE bs+47+i%,ASC(MID$(T%,i%,1))
399 w%1 NEXT
400 JWO END SUB
401 P% REM Gadget Struktur anlegen.
402 g% '-----
403 is SUB FeldDef(bs,nx,x%,y%,b%,h%,f%,a%,T%,i,txt,si,n%) STATIC
404 o%1 ReservSpeicher bs,44% 'Laenge der Struktur
405 m% POKEW bs,nx 'naechstes Gadget
406 x% POKEW bs+4,x% 'xKoordinate linke obere Ecke
407 y% POKEW bs+6,y% 'yKoordinate linke obere Ecke
408 b% POKEW bs+8,b% 'Breite
409 h% POKEW bs+10,h% 'Höhe
410 f% POKEW bs+12,f% 'Gadget-Flags
411 a% POKEW bs+14,a% 'Aktivierung
412 T% POKEW bs+16,T% 'Gadget-Typ
413 i% POKEW bs+18,i% 'Bild oder Hand
414 txt POKEW bs+26,txt 'Text
415 si POKEW bs+34,si 'Zeiger auf Info-Struktur
416 n% POKEW bs+38,n% 'Gadget-Nummer
417 lno END SUB
418 KB REM IntuitionText Struktur anlegen.
419 vp '-----
420 d% SUB IntText(bs,c1%,d1%,x%,y%,T%,nx) STATIC
421 B%1 Groesse=20+LEN(T%)+1 'Strukturlaenge+Textlaenge+Nullbyte
422 y% ReservSpeicher bs,Groesse
423 f% POKE bs,c1% 'Zeichenfarbe
424 m% POKE bs+2,d1% 'Schreibmodus
425 x% POKEW bs+4,x% 'xKoordinate Textanfang
426 y% POKEW bs+6,y% 'yKoordinate Textanfang
427 t% POKEW bs+12,bs+20 'Text
428 m% POKEW bs+16,nx 'naechster Text
429 x% FOR i%=1 TO LEN(T%)
430 m% POKE bs+19+i%,ASC(MID$(T%,i%,1))
431 t%1 NEXT
432 Q%2 END SUB
433 f% REM Raender.
434 f% '-----
435 m% SUB MakeRand(bs,x%,y%,c%,b%,h%) STATIC
436 n%1 ReservSpeicher bs,48% 'Strukturlaenge+Koordinatentabelle
437 t% POKEW bs,x% 'xKoordinate
438 y% POKEW bs+2,y% 'yKoordinate
439 w% POKE bs+4,c% 'Zeichenfarbe
440 m% POKE bs+7,b% '
441 v% POKEW bs+8,bs+16 'xy-Koordinatentabelle
442 a% FOR i%=0 TO 1
443 v% POKEW bs+22+1%*4,h%-1
444 o% POKEW bs+24+1%*4,b%-1
445 n% POKEW bs+32+1%*4,i
446 q% POKEW bs+38+1%*4,h%-1
447 l% POKEW bs+40+1%*4,b%-2
448 J%1 NEXT
449 H%2 END SUB
450 CG PEM Files alphabetisch sortieren
451 w%
452 p% SUB Sor. STATIC
453 g%2 SHARED max%
454 b% Fuelle 6,11,631,36
455 n% druck1 "Ich sortiere ",9,11:druck1 dir.Name$(0),9,19
456 f% druck1 "Bitte ein wenig Geduld.....",9,27
457 b% Lauf%=1 beginn%=1 End%=max% 1
458 L% FOR modus%=0 TO 1
459 V%4 WHILE Lauf%=1

```

```

460 J%6 Lauf%=0
461 Z% FOR i% = beginn% TO End%
462 Q%8 IF UCASE$(dir.Name$(i%)) > UCASE$(dir.Name$(i%+1))
      THEN
463 p% SWAP dir.Name$(i%),dir.Name$(i%+1)
464 z% SWAP dir.Type$(i%), dir.Type$(i%+1)
465 -7 Lauf%=1
      END IF
466 J%8 NEXT i%
467 T%6 beginn%=beginn%+1:Lauf%=0
468 a% FOR i%=End% TO beginn% STEP -1
469 n% IF UCASE$(dir.Name$(i%)) < UCASE$(dir.Name$(i%-1))
      THEN
470 ad% SWAP dir.Name$(i%),dir.Name$(i%-1)
471 s% SWAP dir.Type$(i%), dir.Type$(i%-1)
472 -1 Lauf%=1
      END IF
473 Z%6 NEXT i%
474 t% End%=End%-1
475 a%4 END
476 x%2 NEXT modus%
477 -0 END SUB
480 b% REM Directory vor Disk auflisten
481 b% '-----
482 e% SUB LiesDisk (dir%, max%) STATIC
483 o%2 SHARED fehler%
484 f% lies% = -2
485 x% dir% = dir% + CHR$(0)
486 z% Puffer=252
487 s% Adresse = Lock(SADD(dir%),lies%)
488 w% zaehler%=0
489 z% fehler%=0
490 Q%3 IF Adresse = 0 THEN
491 m%4 Fuelle 6,11,631,36
492 b% druck1 "Inhaltsverzeichnis existiert nicht!",9,19
493 i% FOR i%=0 TO 2000:NEXT
494 x% fehler%=500
495 m% EXIT SUB
496 ap%2 END IF
497 z% opt = 2'16
498 l% info = AllocMem(Puffer,opt)
499 m% IF info = 0 THEN ERROR 7
500 f% erfolg% = Examine$(Adresse,info)
501 b% IF erfolg% = 0 THEN
502 s%4 Fuelle 6,11,631,36
503 a% druck1 "Inhaltsverzeichnis konnte nicht untersucht werden",9,19
504 t% FOR i%=0 TO 2000:NEXT
505 l% fehler%=500
506 f% EXIT SUB
507 V%2 END IF
508 l% WHILE fehler% < > 232
509 g%4 dir.Name = info-8
510 t% FOR schleife% = 0 TO 29
511 i%6 check% = PEEK(dir.Name-schleife%)
512 a% IF check% < > 0 THEN
513 w%8 check% = check%+CHR$(check%)
514 -6 ELSE
515 s%8 schleife%=29
516 t%6 END IF
517 o%4 NEXT schleife%
518 y% dir.Name$(zaehler%) = check%
519 a% check%=""
520 z% typ = PEEKL(info+120)
521 w%8 IF typ < 0 THEN
522 y%6 dir.Type$(zaehler%)="DATEI"
523 o%4 'ELSEIF zaehler% = 0 THEN
524 s%6 ' dir.Type$(zaehler%) = "AKT. DIR."
525 a%4 ELSE
526 w%6 dir.Type$(zaehler%) = "DIR"
527 x%4 END IF
528 b% check%=""
529 d% erfolg%=ExNext$(Adresse,info)
530 t% IF erfolg%=0 THEN
531 b%76 fehler%="0"
532 o% IF fehler% < > 232 THEN
533 s%8 Fuelle 6,11,631,36
534 w% druck1 "Fehler im Inhaltsverzeichnis!",9,19
535 p% FOR i%=0 TO 2000:NEXT
536 p% fehler%=500
537 o% EXIT SUB
538 v%6 END IF
539 -14 ELSE

```



```

540 nn6      zaehler%-zaehler%+1
541 la      IF zaehler%>max% THEN
542 qp8      ferle%-232
543 m56      ELSE
544 a88      ferle%=0
545 jc6      END IF
546 kd4      END IF
547 q42      wEND
548 xm      CALL FreeMem(info,Puffer)
549 br      CALL Unlock(Adresse)
550 z8      max%=zaehler%
551 vx0      END SUB
552 vu      REM Schattenschrift f. Einleitung:
553 pz      -----
554 oN      SUB Schatten (text$,mode%) STATIC
555 AT2      POKEW WINDOW(8)+38,9
556 ln      textlen%=LEN(text$)
557 z8      x%=PEEKW(WINDOW(8)+36)
558 q4      y%=PEEKW(WINDOW(8)+38)
559 p8      COLOR 2,0
560 dL      CALL Move(WINDOW(8),x%+2,y%+2)
561 oN      CALL text(WINDOW(8),SADD(text$),textlen%)
562 hq      CALL SetDrMd(WINDOW(8),0)
563 q8      COLOR 1,0
564 x1      CALL Move(WINDOW(8),x%,y%)
565 g4      CALL text(WINDOW(8),SADD(text$),textlen%)
566 q8      IF mode%=0 THEN
567 v84      PRINT
568 q82      END IF
569 t3      CALL SetDrMd(WINDOW(8),1)
570 EQ0      END SUB
571 4R      REM schnelle Textausgabe im Window:
572 OI      '-----
573 w7      SUB druck1 (txt$,x%,y%) STATIC
574 Vq3      SHARED FensterRP
575 Jm      txtlen%=LEN(txt$)
576 O9      CALL Move(FensterRP,x%,y%+7)
577 h1      CALL text(FensterRP,SADD(txt$),txtlen%)
578 MO0      END SUB
579 Nh      REM Ausgabefelder loeschen
580 4o      -----
581 E1      SUB Fuehle(x1%,y1%,x2%,y2%) STATIC
582 d01      SHARED FensterRP
583 oW      CALL SetAPen(FensterRP,0)
584 Da      RectFill FensterRP,x1%,y1%,x2%,y2%
585 v4      CALL SetAPen(FensterRP,1)
586 UW0      END SUB
587 a8      REM StringInfo Struktur anleger
588 vP      '-----
589 AT      SUB StrInfo(stringpuffer,buffer) STATIC
590 lv1      SHARED EditGroesse
591 E1      max%=15
592 Mx      IF LEN(buffer) > max% THEN buffer=LEFT$(buffer,max%)
593 oU      IF (max% AND 1) THEN max%=max%+1
594 lQ      EditGroesse=36+2*(max%+4)
595 pY      stringpuffer=AllocMem(EditGroesse,65536)
596 mK      IF stringpuffer=0 THEN ERROR 7
597 Tk      POKEW stringpuffer,stringpuffer+36
598 na      POKEW stringpuffer+10,16
599 3R      IF buffer<>" THEN
600 s13      FOR i%-1 TO LEN(buffer)
601 t83      POKE stringpuffer+35+i%,ASC(MID$(buffer,i%,1))
602 na3      NEXT
603 FY1      END IF
604 mo0      END SUB
605 m3      REM editierten String aus Speicher Leser
606 YP      '-----
607 DV      SUB LiesEdit (stringpuffer,ret%) STATIC
608 JD2      SHARED EditGroesse
609 p1      puffer.string=PEEKL(stringpuffer)
610 l1      ret%=""
611 sS      WHILE check% > 0
612 JF4      check%=(PEEK(puffer.string-i%))
613 xT      ret%=ret%+CHR$(check%)
614 vx      i%=i%+1
615 MA2      wEND
616 M8      ret%=LEFT$(ret%,LEN(ret%)-1)
617 q5      CALL FreeMem(stringpuffer>EditGroesse)
618 O20      END SUB
619 ga      REM linke Mausstecke ??
620 dx      '-----
621 z0      SUB Maus STATT
622 pg1      Jimmy-MOUSE

```

```

623 Gg      WHILE MOUSE(0) = 0
624 V3      wFNE
625 T90      END SUB
626 hS      REM Druckroutine
627 v2      -----
628 CQ      SUB Etikett STATIC
629 z1      REM ***
630 AH2      SHARED FensterAdresse,DruckFrageTxt,JaTxt,NeinTxt,Druck%
631 EI      Antwort=AutoRequest(FensterAdresse,DruckFrageTxt,JaTxt,Ne
        in.txt,0%,0%,300,50
632 l1      IF Antwort < 1 THEN
633 SC4      GOTO aus
634 AJ2      wND IF
635 J3      anzahl%=18
636 wN      esc%=CHR$(27) : init$=esc%+"@"
637 iz      fet$=init$+esc%+"B"+esc%+"w"+CHR$(1
638 ax      absta=d$+esc%+"A"+CHR$(6
639 iQ      k.e!$=esc%+"SC"
640 Ab      eng%=CHR$(15)
641 p8      elite$=esc%+"X"
642 pJ      Ende$=esc%+"A"+CHR$(13)+esc%+"2"
643 Ap      titel2$=init$+esc%+"2"
644 p2      titel2$=LEFT$(Druck$(0),13)
645 LX      OPEN "par " FOR APPEND AS #1
646 2R4      PRINT #1,elit$,abstand$ : 'IBM-MODE +esc%+"2",
647 BI      PRINT #1,elit$,eng$,
648 Qb      PRINT #1,SPC(29);"KEIN Schreibschutz"
649 Wf      PRINT #1,
650 aJ      PRINT #1,SPC(34);"Schreibschutz"
651 Yn      PRINT #1,
652 u9      einrueck=INT((48-LEN(titel2$))/2) REM ** Titel mittig
        **
653 Ga      justify=24-INT(LEN(titel2$)/2)-4
654 pN      PRINT #1,SPC(einrueck)+titel2$
655 l1      einrueck=INT((14-LEN(titel2$))/2) REM ** Titel mittl
        2 **
656 e2      PRINT #1,fe-$
657 6W      PRINT #1,SPC(einrueck);titel2$
658 Ab      PRINT #1,esc%+"AC",
659 yI      PRINT #1,init$,
660 fx      IF Druck% > 19 THEN
661 4b6      anzahl% = 32
662 f9      PRINT #1,abstand$, 'IBM-MODE +esc%+"2";
663 aC      PRINT #1,elite$,
664 eX4      END IF
665 ut      REM ** Rarnen **
666 JC      PRINT #1,elite$;CHR$(32); 'IBM-MODE ch-$ 201)
667 Er      FOR i=1 TO 15
668 w26      PRINT #1,CHR$(45), : 'IBM (205)
669 EU4      NEXT i
670 dv      PRINT #1,CHR$(32), : 'IBM (203)
671 Iv      FOR i=1 TO 15
672 O66      PRINT #1,CHR$(45), : 'IBM (205)
673 w14      NEXT
674 M4      PRINT #1,CHR$(32) : 'IBM (187)
675 TH      FOR i=1 TO anzahl% STEP 2
676 X06      v1=INT((15-LEN(Druck$(i))/2)
677 vD      n1=INT(14-LEN(Druck$(i+1))/2)+5
678 OQ      v2=INT((15-LEN(Druck$(i+1))))/2)
679 wo      n2=INT((15-LEN(Druck$(i+1))))/2)+5)
680 O4      PRINT #1,CHR$(124);SPC(v1);Druck$(i);SPC(n1);CHR$(124
        ); : 'IBM (186)
681 u3      PRINT #1,SPC(v2);Druck$(i+1);SPC(n2);CHR$(124)
        : 'IBM (186)
682 Rh4      NEXT i
683 Y2      PRINT #1,CHR$(32); : 'IBM (200)
684 V8      FOR i=1 TO 15
685 DJ6      PRINT #1,CHR$(45); : 'IBM (205)
686 V24      NEXT i
687 oA      PRINT #1,CHR$(32); : 'IBM (202)
688 ZC      FOR i=1 TO 15
689 Hn6      PRINT #1,CHR$(45); : 'IBM (205)
690 Zp4      NEXT i
691 JN      PRINT #1,CHR$(32) : 'IBM (188)
692 LF      PRINT #1,klein$;
693 YX      PRINT #1,Ende$
694 QD      CLOSE #1
695 s3      aus
696 q7      FOR i%=0 TO 32
697 DB5      Druck$(i%)=""
698 LQ4      NEXT
699 J10      END SUB
(C, 1990 4eT

```

Listing. »DiskEtI«
(Schluß)

und läuft und läuft und läuft und

Schluß mit den Befehlen »Echo« und »Type«, es lebe die Laufschrift! Dabei haben Sie vollen Einfluß auf den scrollenden Text. Dieser kann beliebig plaziert, sowie in unterschiedlicher Breite und mit variabler Geschwindigkeit dargestellt werden.

Unser Tool »hscroll« (Listing 1) ist in Aztec-C geschrieben. Geben Sie das Listing bitte mit dem Checksummer ein. Zur Compilation tippen Sie einfach »cc +l hscroll.c« gefolgt von »in hscroll.o -lc32« ein. Danach wird es am besten in den C-Ordner kopiert, so daß es jederzeit zur Verfügung steht. Folgende Optionen stehen zur Verfügung, die alle mit einem Leerzeichen getrennt hinter »hscroll« eingegeben werden:

Parameter für hscroll

- ix: x ist die Position auf der x-Achse für die Laufschrift (Werte von 0 bis 639)
- ty: y gibt die y-Position an (Werte von 1 bis 256 beim PAL-Amiga)
- wn: setzt die Breite des Ausgabefensters auf n Pixel. Das Fenster kann maximal 640 Punkte breit sein, wenn Sie Null bei -1 angeben.
- sn: n legt die Geschwindigkeit fest, die Werte von 1,2,4 oder 8 annehmen kann.
- rn: n entspricht hier der Anzahl der Wiederholungen. Bei n=1 läuft Ihr Text nur einmal durch

Der eigentliche Text der Laufschrift wird mit einem Editor geschrieben und als ASCII-File abgespeichert. Er kann dann ebenfalls als Parameter hinter »hscroll« übergeben werden. Beispiel:

```
hscroll Dateiname -10 -t100 -w200
```

Dieser Aufruf schickt eine Laufschrift am linken Rand des Screens (etwa in der Mitte) über eine Breite von 200 Bildpunkten. Eventuelle Returns innerhalb des Ausgabe-Textes werden durch Leerzeichen ersetzt. Dies ermöglicht ungeahnte Effekte: Lassen Sie Ihre eigenen Fehlermeldungen anzeigen,

beim Compilieren den Source-Code durchlaufen oder geben Sie Directories in Laufschrift aus.

Haben Sie eine der Optionen nicht verwendet, so werden Default-Werte benutzt, von denen Sie allerdings nur den für »-r« kennen müssen. Ohne »-r« gibt es eine annähernd unbegrenzte Wiederholungszahl, nämlich genau $2^{32}-1$. Dies sind immerhin etwas über 4 Milliarden Wiederholungen; für die meisten Anwendungen dürfte das ausreichen. Dies heißt aber nicht, daß Sie Ihre Laufschrift jetzt eine Woche lang betrachten müssen, sie kann auch mit der Maus weggeklickt werden.

Mehr Effekt mit »Run«

Sie können »hscroll« auch problemlos als Background-Task starten, so daß die Laufschrift im Hintergrund ihrer Wege geht. Dies kann beispielsweise sinnvoll sein, wenn ein längeres Programm geladen werden soll und währenddessen eine Laufschrift das Ganze ein wenig schmücken könnte. Installieren Sie diese also zuerst mit »run hscroll« und den entsprechenden Parametern (am besten in einer Batch-Datei) und erst danach kommt der eigentliche Befehl zum Laden. Dabei kann es vorkommen, daß der Ladevorgang geringfügig langsamer wird (je nach Größe der Ausgabe). Soll die Laufschrift noch kurz vor Beendigung des Ladens verschwinden, können Sie die Option »-r« verwenden.

Es gibt fünf Dinge, die Sie bei der Benutzung dieses Tools wissen sollten:

- Das Scrolling ist nicht unbedingt immer stufenlos, wenn die Laufschrift ganz oben im Bild-

schirm geöffnet wird. Das hängt damit zusammen, daß der Task immer auf die vertikale Austastlücke wartet. Dies ist - vereinfacht gesagt - der Moment, in dem der Elektronenstrahl sich in Zeile 0 befindet. Je weiter oben sich nun die Laufschrift befindet, desto weniger Zeit verbleibt, den Window-»Rast-Port« zu scrollen, ohne daß zugleich das Bild dargestellt wird. Wenn hierfür nicht genügend Zeit ist, flimmert die Textdarstellung. Der Einfachheit halber wurde auch auf das sogenannte »Double Buffering« verzichtet.

- Sind mit »run hscroll« mehrere Tasks aufgerufen worden, so kann es leicht passieren, daß Ihr CLI Task sozusagen »out of CPU« gerät, also keine Rechenzeit mehr bekommt. Das liegt daran, daß die Laufschrift-Tasks bei Aufruf die höchstmögliche Priorität bekommen, um ein Ruckeln zu verhindern. Man mache sich klar, daß die vertikale Austastlücke etwa 50 mal pro Sekunde auftritt und »hscroll« ebensooft den Prozessor benötigt.

- Es können nur 8 x 8-Punkt Zeichensätze verwendet werden. Wer also in seinem CLI andere benutzen möchte, der sollte erst einmal diese Einschränkung überprüfen. Aus diesem Grunde beträgt die Höhe des geöffneten Scroll-Fensters auch immer 8 Pixel.

- Laufschriften sind nichts anderes als kleine Fenster. Sie können von anderen Fenstern überlappt werden und dann einfach verschwinden.

- Jedes Programm, das längere Blitter-Operationen veranlaßt, ist ein potentieller Störfaktor bezüglich des Filescrollings. Schließlich benutzt auch »hscroll« den Blitter zum Scrol-

ling. Irgendwann aber darf jeder Task mit auch noch so kleiner Priorität mal den Blitter verwenden, was ein leichtes Ruckeln zur Folge hat. Dies kann nicht umgangen werden, wenn man sich konsequent an die Multitasking-Konventionen halten will.

Zur Beschleunigung der Textausgabe gibt es auf dem Amiga einige nützliche Utilities, unter anderem auch »Blitz-Fonts!« und »FastFonts«, das sich auf der neuen Workbench 1.3 befindet. Diese wirken sich auch günstig auf unsere laufenden Texte aus. Sie sind also unbedingt zu empfehlen, vor allem wegen der geringeren Rechenzeit.

Um dem Laufschrift-Task ein möglichst ruckelfreies Dasein zu beschaffen, wird mittels »SetTaskPri(FindTask(0),127);« die Priorität desselben auf den größtmöglichen Wert gesetzt. »FindTask(0)« durchsucht die Liste der gerade ablaufenden Tasks und übergibt den entsprechenden Zeiger auf die TaskStruktur. Nach Beendigung der Hauptschleife wird die Priorität dann auf einen normalen Wert zurückgesetzt. Es werden die üblichen Libraries geöffnet, die für die Verwaltung von Fenstern und Grafik notwendig sind. Um den einzulesenen ScrollText dynamisch verwalten zu können, muß dessen Länge bekannt sein.

Besser geht's mit BlitzFonts!

Es wird daher Speicher für die Daten des »FileInfoBlocks« reserviert, aus dem man nach Untersuchung des Files die Länge desselben entnehmen kann (»FileInfoBlock->fib_Size«).

Laufschriften sind eine Methode, Informationen auszugeben. Sollten Sie vor deren Programmierung zurückschrecken, dann brauchen Sie dieses Utility. Mit »hscroll« bringen Sie Schwung in die Langeweile des CLI.

und läuft und läuft und läuft und

Sozusagen als Unterlage für die Laufschrift wird ein einfaches Fenster »ohne alles« geöffnet, welches im übrigen genau ein Gadget zum Wegklicken enthält.

Die Hauptschleife wartet zunächst mit »WaitTOF()« auf die nächste Austastlücke und scrollt dann den gesamten »RastPort« des Fensters um p Pixels (je nach eingestellter Geschwindigkeit) nach links. Mit Hilfe der Text-Funktion wird dann der neu hinzugekommene Teil neu geschrieben. Das ist in diesem Fall äußerst einfach, weil Operationen im RastPort grundsätzlich nicht über den Rand hinaus gehen, also nichts außerhalb unseres Fensters verändern. Eventuelle Reste werden einfach abgeschnitten.

Smooth mit WaitTOF()

Die Hauptschleife wird solange durchlaufen, bis entweder die Variable für die Wiederholungen abgelaufen ist oder eine Nachricht am Port des Fensters anliegt. Auch dieser Fall ist leicht zu handhaben, da wir nur ein Gadget zur Verfügung haben, das den IDCMP des Fensters ansprechen kann; damit können wir uns eine Auswertung der Nachricht sparen.

Möglicherweise stört Sie die Beschränkung von »hscroll« auf eine Zeile? Die Lösung heißt

»vscroll« (Listing 2). Dieses Programm ist in den wesentlichen Punkten genauso aufgebaut. Das Textfile wird jetzt allerdings nach der längsten enthaltenen Zeile abgesucht. Diese bestimmt die Breite des Ausgabe Fensters. Falls der rechte Rand des Screens überschritten wird, bricht das Programm vorher ab. Selbstverständlich arbeitet »vscroll« auch mit exotischen Formaten wie der 704 Pixel breiten Workbench oder dem PD-Programm »VScreen-Handler« zusammen.

Bitte beachten Sie den Listingausschnitt von Zeile 163 bis Zeile 166. Hier wird dem Betriebssystem suggeriert, es würde nur eine Bitplane bestehen. Durch diesen kleinen Trick erreichen wir eine geringere Belastung des Blitters für die Scroll-Operationen. Vor Beenden des Programms werden die Bitplanes wieder auf Normalwerte gesetzt.

»vscroll« können Sie mit dem Aztec (»cc vscroll.c« gefolgt von »ln vscroll.o -lc«) oder mit dem Lattice (»lc -w vscroll«) compilieren.

Vielleicht fallen Ihnen selbst noch einige Verbesserungen ein, wie zum Beispiel verschiedene Schriftarten, per Tastendruck verschiebbares Fenster oder ähnliches? Wir wünschen Ihnen auf jeden Fall viel Spaß beim Experimentieren mit diesem Utility. *Christian Wolf/kn*

Programmname:	hscroll
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Aztek-C
Compiler:	Version 3.6
Aufrufe:	cc +l hscroll.c ln hscroll.o -lc32

```

Programmautor (c) Christian Wolf
-----
1 0f0 #include exec/types.h
2 xV #include intuition/intuition.h
3 Lt #include intuition/intuitionbase.h
4 dl #include exec/exec.h
5 tP #include exec/execbase.h
6 'd #include libraries/dos.h
7 vS #include libraries/dosextens.h
8 xJ #include graphics/gfxmacro.h
9 Ho #include functions.h
10 Rn #define NO_OPIEP NULL
11 Yv #define NO_CRT NULL
12 Fe /* Gadget zum Abklicken */
13 EN struct Gadget *hscrollGadget = NULL;
14 5vU

```

```

15 '00 * Einfaches Fenster ohne alles. */
16 '09 struct NewWindow *hscrollWindow =
17 '07 0,230,640,640,1,
18 '02 GADGETUP,
19 '06 0,0,0,0,0,
20 '04 0,0,0,0,0,
21 '04 0,0,0,0,0,
22 '02 0,0,0,0,0,
23 '02 0,0,0,0,0,
24 '04 0,0,0,0,0,
25 '04 0,0,0,0,0,
26 '04 0,0,0,0,0,
27 '04 0,0,0,0,0,
28 '04 0,0,0,0,0,
29 '04 0,0,0,0,0,
30 '04 0,0,0,0,0,
31 '04 0,0,0,0,0,
32 '04 0,0,0,0,0,
33 '04 0,0,0,0,0,
34 '04 0,0,0,0,0,
35 '04 0,0,0,0,0,
36 '04 0,0,0,0,0,
37 '04 0,0,0,0,0,
38 '04 0,0,0,0,0,
39 '04 0,0,0,0,0,
40 '04 0,0,0,0,0,
41 '04 0,0,0,0,0,
42 '04 0,0,0,0,0,
43 '04 0,0,0,0,0,
44 '04 0,0,0,0,0,
45 '04 0,0,0,0,0,
46 '04 0,0,0,0,0,
47 '04 0,0,0,0,0,
48 '04 0,0,0,0,0,
49 '04 0,0,0,0,0,
50 '04 0,0,0,0,0,
51 '04 0,0,0,0,0,
52 '04 0,0,0,0,0,
53 '04 0,0,0,0,0,
54 '04 0,0,0,0,0,
55 '04 0,0,0,0,0,
56 '04 0,0,0,0,0,
57 '04 0,0,0,0,0,
58 '04 0,0,0,0,0,
59 '04 0,0,0,0,0,
60 '04 0,0,0,0,0,
61 '04 0,0,0,0,0,
62 '04 0,0,0,0,0,
63 '04 0,0,0,0,0,
64 '04 0,0,0,0,0,
65 '04 0,0,0,0,0,
66 '04 0,0,0,0,0,
67 '04 0,0,0,0,0,
68 '04 0,0,0,0,0,
69 '04 0,0,0,0,0,
70 '04 0,0,0,0,0,
71 '04 0,0,0,0,0,
72 '04 0,0,0,0,0,
73 '04 0,0,0,0,0,
74 '04 0,0,0,0,0,
75 '04 0,0,0,0,0,
76 '04 0,0,0,0,0,
77 '04 0,0,0,0,0,

```

Listing 1. »hscroll.c« zaubert Ihnen eine horizontale Laufschrift auf den Bildschirm

```

80 7u9      else
81 FiD      {
82 YKE      /* TextFile laden... */
83 Wj      if (ThisFile=Open(*argv,MODE_OLDFILE))
84 lII      {
85 qMj      MyLock=Lock(*argv,ACCESS_READ);
86 xi      Examine(MyLock,FileInfoBlock);
87 2d      Size=FileInfoBlock->fib_Size;
88 iW      if (MyPtr=ScrollText=AllocMem(Size,MEMF
PUBLIC))
      {
89 NqO      Read(ThisFile,ScrollText,Size);
90 TWP      MyPtr[Size]=0;
91 dj      /* eventuelle returns entfernen...
92 QR      */
          while (*MyPtr)
          {
93 Db      if (*MyPtr == 0x0a) *MyPtr
94 SvW      ++;
95 2WX      else MyPtr++;
          }
          Unlock MyLock ,
          Close(ThisFile),
          else ScrollText=DefaultText,
          {
100 e7O      sl=strlen ScrollText';
101 QdJ      }
102 e9I      ,
103 hPE      ,
104 gBD      ,
105 0o9      argv++,
106 iDS      ,
107 Q5i      /* Libraries Öffnen... */
108 DE      IntuitionBase=(struct IntuitionBase *)OpenLibrary("intuiti
on.library",0);
109 mQ      GfxBase=(struct GfxBase *)OpenLibrary("graphics.library",0
);
110 Tr      /* Werte für Fenster und Gadget initialisieren */
111 xu      NewScrollWindow.LeftEdge=1;
112 li      NewScrollWindow.TopEdge=1;
113 wp      NewScrollWindow.Width=w;
114 Wz      ClickGadget.LeftEdge=0;
115 LK      ClickGadget.TopEdge=0;
116 vV      ClickGadget.Width=w;
117 Bh      /* Fenster öffnen */
118 Sk      if (!{ScrollWindow=OpenWindow(&NewScrollWindow)}) exit(0);
119 gF      /* Rasterport holen */
120 To      ScrollRastPort=ScrollWindow->RPort;
121 vv      SetAPen(ScrollRastPort,3);
122 mk      SetBPen(ScrollRastPort,0);
123 2q      SetDrMd(ScrollRastPort,JAM1);
124 dx      /* Hauptschleife -> Scrollen, solange ri=0 oder UserPort
leer */
125 iX      while (!GetMsg(ScrollWindow->UserPort) && r)
126 ,R2
127 5J3      /* Auf den VBL warten */
128 lj      WaitTOP();
129 MR      /* Grafik um s Pixels verschieben... */
130 Sv      ScrollRastPort,ScrollRastPort,s,0,C,0,w-1,7),
131 S7      Move(ScrollRastPort,w-s-p,6);
132 KC      if (sl <= c)
133 5X7      {
134 wB8          c=0;
135 Te          r--;
136 Ch7
137 eV3      /* Nachfolgenden Text reinschreiben */
138 Mw      Text(ScrollRastPort,&ScrollText[c],1);
139 X4      p+=s;
140 9w      if (p > 8-s)
141 Dg7          p=0;
142 Uw8          c++;
143 tl
144 Kp7
145 lq2
146 p11      /* Alles schließen... */
147 ER      CloseWindow(ScrollWindow);
148 fE      CloseLibrary(IntuitionBase);
149 pL      CloseLibrary(GfxBase);
150 w8      if (ScrollText!=DefaultText) FreeMem(ScrollText,sl);
151 s9      if (FileInfoBlock) FreeMem(FileInfoBlock,sizeof(struct Fil
eInfoBlock));
152 r9      SetTaskPri(FindTask(0),0);
153 TyQ      }
(C) 1989 M&T

```

Listing 1. (Schluß)

Programmname: vscroll

Computer: A500, A1000, A2000 mit Kickstart 1.2, 1.3

Sprache: C

Compiler: Aztec oder Lattice

Aufrufe: siehe Text

Programmautor: Christian Wolf

```

1 i60 #include <intuition/intuitionbase.h>
2 Qn #include <libraries/dos.h>
3 AP #include <exec/memory.h>
4 dx #include <graphics/gfxbase.h>
5 57 #ifdef LATTICE
6 2d #include <proto/all.h>
7 f6 #else
8 vV #include <functions.h>
9 wX #endif
10 vV #define fib FileInfoBlock
11 2d #define fibsize (long)sizeof(struct fib)
12 15 #define NOBORDER NULL
13 aL #define NOTEXT NULL
14 vV /* Gadget zum Abklicken... */
15 5s struct Gadget ClickGadget = { NULL,0,0,0,0,GADGNONE,RELVER
IFY,
16 7xU      BOOLGADGET,NOBORDER,0,NOTEXT,
0,0,NULL,0};
17 v80 /* Einfaches Fenster ohne alles... */
18 BB struct NewWindow NewScrollWindow = {
19 2,7      0,230,640,8,0,1,
20 28      GADGETUP|GADGETUP,
21 26      BORDERLESS,
22 rL      &ClickGadget,NULL,
23 20      NULL,
24 24      NULL,NULL,
25 22      640,200,640,200,
26 71      WBENCHSCREEN};
27 v30 /* Das übliche... */
28 vQ struct IntuitionBase *IntuitionBase;
29 Fs struct GfxBase *GfxBase;
30 8s struct Table { char *Line;
31 oAF      long Len; };
32 2f0 struct Table DefaultText[] = { "      VScroll 1.0
",37,
33 22V      "      (c) 1989 by Markt und Te
chnik",37,
34 7n      "      written by Christian W
olf",37,
35 24      "      currently resident at
",37,
36 72      "      Spandauer Str. 104L
",37,
37 71      "      1 Berlin 20
",37,
38 72      "      Tel.: 030/3660450
",37,
39 71      "      Usage: vscroll textfile [
options]",37,
40 71      "      Use -w to modify window's
width",37,
41 71      "      Use -s number of jiffies
1,2,4,8",37,
42 71      "      Use -l and -t to position
window",37,
43 22      "      Use -r for no. of repetit
ions",37,
44 71      "      Use -h for a new height o
f window",37,
45 71      "
",37,
46 71      "
",37,
47 22      "
",37,
48 71      "
",37,0};
49 2f0 struct BitMap ScrollBitmap;
50 71 void main (argc,argv)
51 301      int *argv;
52 71      int argc;
53 71

```



```

54 Otl struct Table *ScrollText=DefaultText,*cl=DefaultText;
55 nJ char *MyPtr,*LoadText=0;
56 fV struct Window *ScrollWindow;
57 Gg struct RastPort *ScrollRastPort;
58 9W long l=160,t=60,w=378,s=1,p=0,c=0,h=100,s1;
59 8S long r=0xffffffff,Size,Lines;
60 a6 struct fib *fib;
61 Uy struct FileHandle *ThisFile;
62 We struct FileLock *MyLock;
63 V1 int i;
64 Qo /* Damit es nicht so ruckelt -> höchste Taskpriorität */
65 Py SetTaskPri(PindTask(OL),127L);
66 bB /* parameter liste durchgehen */
67 7W argc--;
68 PD argv++;
69 LQ while (argc--)
70 AXB {
71 7a9     if (**argv=="-")
72 62D     {
73 1FE         (*argv)++;
74 nO         switch( *argv++)
75 9oL         {
76 16M             case 'h':
77 Md                 h=atoi(*argv);
78 BK                 break;
79 xP             case 'l':
80 x6                 l=atoi(*argv);
81 EN                 break;
82 9a             case 't':
83 qJ                 t=atoi(*argv);
84 1Q             case 's':
85 Ox                 s=atoi(*argv);
86 rW                 break;
87 KT             case 'r':
88 Qw                 r=atoi(*argv);
89 sJ                 break;
90 NW             case 'p':
91 TyL                 p=atoi(*argv);
92 UzD             case 'c':
93 K79                 c=atoi(*argv);
94 SvD             case 'h':
95 6SE                 h=atoi(*argv);
96 UxI             case 'w':
97 nZJ                 w=atoi(*argv);
98 8f             case 's1':
99 XON                 s1=atoi(*argv);
100 ERO             case 's':
101 R9                 s=atoi(*argv);
102 a3S             case 'p':
103 pJT                 p=atoi(*argv);
104 Nj             case 'c':
105 zO                 c=atoi(*argv);
106 T5             case 'h':
107 Hm             case 'w':
108 g9Y             case 's1':
109 vR2             case 's':
110 15                 s=atoi(*argv);
111 NY                 break;
112 kDe             case 'r':
113 Jzf             case 'r':
114 qLe             case 'r':
115 OGZ             case 'r':
116 VS             case 'r':
117 pJo             case 'r':
118 Bkd             case 'r':
119 rKf             case 'r':
120 Rbg             case 'r':
121 ve             case 'r':
122 yTf             case 'r':
123 Qad             case 'r':

```

```

124 OVe             case 'r':
125 11Z             case 'r':
126 VV             case 'r':
127 aSe             case 'r':
128 WJf             case 'r':
129 5ae             case 'r':
130 76Z             case 'r':
131 2K             case 'r':
132 8dY             case 'r':
133 v9T             case 'r':
134 J8             case 'r':
135 8gS             case 'r':
136 zOO             case 'r':
137 ag             case 'r':
138 EJN             case 'r':
139 PkI             case 'r':
140 Q1D             case 'r':
141 aO9             case 'r':
142 In8             case 'r':
143 Of1 /* Libraries öffnen... */
144 VE IntuitionBase=(struct IntuitionBase *)OpenLibrary("intuiti
on.library",0L);
145 J7 GfxBase=(struct GfxBase *)OpenLibrary("graphics.library",0
L);
146 JR /* Werte für Fenster und Gadget initialisieren */
147 1a NewScrollWindow->LeftEdge=0;
148 4b NewScrollWindow->TopEdge=0;
149 1a NewScrollWindow->Width=w;
150 rI NewScrollWindow->Height=h;
151 7a ClickGadget->LeftEdge=0;
152 wv ClickGadget->TopEdge=0;
153 oZ ClickGadget->Height=h;
154 xI ClickGadget->Width=w;
155 nJ /* Fenster öffnen */
156 aX if (ScrollWindow=(struct Window *)OpenWindow(&NewScrollWin
dow))
157 Tv5 {
158 Js6     /* Rasterport holen */
159 GR     ScrollRastPort=ScrollWindow->RPort;
160 VM     SetAPen(ScrollRastPort,1L);
161 WN     SetBPen(ScrollRastPort,0L);
162 CT     SetDrMd(ScrollRastPort,JAM1);
163 Np     /* BitMap kopieren */
164 3k     ScrollBitMap=&ScrollRastPort->BitMap;
165 QW     ScrollRastPort->BitMap=&ScrollBitMap;
166 7Q     ScrollBitMap->Depth=1;
167 oE     h+=6;
168 BE     while (!GetMsg(ScrollWindow->UserPort) && r)
169 f87     {
170 OPe         WaitTOP();
171 C8         ScrollRaster(ScrollRastPort,0L,a,0L,0L,w-1,h-1);
172 cs         Move(ScrollRastPort,0L,h-s-p);
173 k6         Text(ScrollRastPort,cl->Line,cl->Len);
174 6d         p++;
175 1V         if (p > s-a)
176 aFC         {
177 JVD             p=0;
178 vq             cl++;
179 2O             if (cl->Line)
180 qJH             {
181 q8I                 cl=ScrollText;
182 EP                 r--;
183 xSH             }
184 yTC             }
185 aU7             }
186 Xa6         ScrollBitMap->Depth=2;
187 jM         /* alles schließen */
188 t6         CloseWindow(ScrollWindow);
189 JY9             }
190 QRI CloseLibrary((struct Library *)IntuitionBase);
191 AQ CloseLibrary((struct Library *)GfxBase);
192 8y if (ScrollText!=DefaultText) FreeMem(ScrollText,Lines<<3
);
193 T9 if (LoadText) FreeMem(LoadText,Size);
194 7G SetTaskPri(PindTask(OL),OL);
195 9e0 }
(C) 1989 M&T

```

Listing 2. Mit »vscroll« bewegen Sie ein ganzes Textfile vertikal über den Screen

Eine Dateiverwaltung in Basic? »Langsam und unkomfortabel«? Weit gefehlt: Die »Uni-Datei« ist schnell und besticht mit einer Benutzerführung, deren Komfort so manchen C-Programmierer vor Neid erblassen läßt.

Von Rainer Karwies
und Andreas Lietz

Kennen Sie den »Basic-Effekt«? Da benutzt man ein selbstgeschriebenes oder abgetipptes Programm in Amiga-Basic und ermittelt von den C-, Assembler- und Modula-Freunden nur ein Lächeln, wenn man ihnen das Programm vorführt. Mit der »Uni-Datei« wird das anders. Dieses Programm benutzt die Fähigkeiten des Amiga zur komfortablen Benutzerführung so ausgiebig, daß man fast vergessen könnte, in welcher Sprache es geschrieben ist! Der große Vorteil dieses Programmes ist, daß Sie das Programm jederzeit an Ihre Bedürfnisse anpassen können, wenn Sie über Basic-Kenntnisse verfügen.

Die Uni-Datei ist eine sequentielle Dateiverwaltung. Sie kann Datensätze mit maximal 18 Datenfeldern verwalten; die Anzahl der Datensätze hängt vom verfügbaren Speicherplatz und von der Anzahl der Felder ab. Ist die Feldanzahl höher, wird die Anzahl der Datensätze geringer und umgekehrt.

Das Programm ist darauf ausgelegt, dem Benutzer einen möglichst großen Komfort zu bieten. Mit Hilfe der Maus und kassettenrecorder-ähnlichen Tasten (Bild 1 unten) wählen Sie einzelne Datensätze schnell an. Es werden viele Menüs verwendet, mit deren Hilfe alle Funktionen des Programmes aufgerufen werden. In einer Datei kann nach einem bestimmten Feld gesucht und sortiert werden; die Daten lassen sich automatisch auf Etiketten oder als Liste drucken. Es ist jederzeit möglich, die Feldnamen zu ändern. Mit einem Hilfsprogramm ändern Sie sogar die Anzahl der Felder oder deren Reihenfolge, ohne daß dabei Daten verlorengehen.

Der ganze Komfort hat nur einen Preis: Das Programm ist zu lang zum Abtippen. Deshalb wird es auf der Programmservice-Diskette zu diesem Heft veröffentlicht. Auf dieser Diskette finden Sie die beiden Programmversionen »Uni_0.5MB« und »Uni_1MB« sowie die Ladeprogramme »Lader-Uni_0.5MB« und »Lader-Uni_1MB«. Die erste Version ist für Besitzer eines Amiga mit 512 KByte Speicher gedacht; wer mehr Speicher hat, verwendet die zweite Version. Außerdem ist auf der Diskette das Hilfsprogramm »UniUtil«, mit dem Sie bei vorhandenen Dateien die Feldanzahl ändern und die Reihenfolge der Felder vertauschen können.

Um mit der Uni-Datei zu arbeiten, sollten Sie sich erst einmal eine Arbeitsdiskette formatieren. Öffnen Sie dann die Schublade »Uni-Datei« auf der Programmservice-Diskette. Haben Sie mehr als 512 KByte Speicher, so kopieren Sie nun alles außer »Lader-Uni_0.5MB« und »Uni_0.5MB« auf ihre Diskette. Wenn Sie mit 512 KByte Speicher arbeiten, kopieren Sie alles außer »Lader-Uni_1MB«, »Uni-Datei 1MB«, »exec.bmap« und »dos.bmap« auf Ihre Ar-

Bild 1. Die Uni-Datei in Aktion. Den Aufbau Ihrer Datei bestimmen Sie selbst – hier ein Beispiel.

Dateiname: Df1:Daten/Hardcopy System: 2637000 Basic: 39131

Datensatz Nr. 2 von 2 Datum 12.08.1988 17:47:46

Herr/Frau	Herrn
Vorname	Peter
Name	Pan
Strasse	Ruberzahlstr. 127
PLZ	8000
Ort	München 12
Bezirk	Südbayern
Vorwahl	089
Telefon	1234567
Bemerkung 1	Guter Kunde; tendiert zu unfreundlichem Auftreten
Bemerkung 2	Gute Beziehungen zu Schweden
Kundennummer	1254653-236
Auftragsnr.	12-45/78
Let. Auftrag	12.6.89
Kredit	3000,-
Rabatt	0 %
Skonto	3 %
Satz OK	Ja

Navigationstasten: < << <<< < > >> >>> >

Ja Nein

beitsdiskette. Für alle Versionen muß schließlich noch Amiga-Basic auf die Diskette kopiert werden – Sie finden es auf Ihrer Extras-Diskette. Nun haben Sie eine Arbeitsdiskette, auf der viel Platz für Daten zur Verfügung steht. Bevor Sie mit der Uni-Datei arbeiten, muß sie noch für Ihre Wünsche angepaßt werden. Dazu laden Sie das Programm mit einem Doppelklick auf »Lader-Uni_1MB« bzw. »La-

kann, hängt davon ab, welche Programme sonst noch in Ihrem Rechner Speicher belegen. Haben Sie mehr als 1 MByte Speicher zur Verfügung, können Sie den Wert natürlich kräftig hochsetzen.

Einige Zeilen weiter unten sehen Sie die Zuweisung »Max=250«. Diese Zahl ist die maximale Zahl der Datensätze und für normale Anwendungen völlig ausreichend. Haben Sie

aber viel Speicher oder Datensätze mit sehr wenigen Feldern, so können Sie diese Zahl hochsetzen – meldet AmigaBas c daraufhin »Out of Memory«, so müssen Sie »Max« wieder reduzieren. Umgekehrt sollten Sie »Max« niedriger festlegen, wenn Ihre Datensätze 18 Felder haben und Sie mit 512 KByte Speicher auskommen müssen.

Nun folgt die Zuweisung

DName\$ = »Adressen«

Hier setzen Sie statt »Adressen« den Namen der Datei ein, die Sie am häufigsten nutzen. Dieser Name wird beim Laden von Dateien vom Programm vorgegeben (kann aber natürlich jederzeit überschrieben werden). Schließlich folgt die Anweisung

Pfad\$ = »df1:Daten«

An Dateianfang springen Fünf Datensätze zurück



50 Datensätze zurück

Bild 3. So verwenden Sie die

verselle waltung

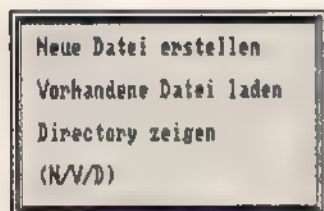


Bild 2. Diese blinkende Box erscheint beim Start der Uni-Datei

Hier wird angegeben, wo die Daten vom Programm abgelegt werden. Bitte stellen Sie diese Variable entsprechend Ihren Wünschen ein. Speichern Sie jetzt das Programm und starten Sie es erneut.

Dazu klicken Sie das jeweilige Lader-Programm an und warten, bis der Bildschirm wie in Bild 2 aussieht. Da Sie bisher noch keine Datei erzeugt haben, tippen Sie »n«, um eine neue Datei zu erstellen. Nun müssen Sie die Anzahl der Datenfelder und die Bezeichnung der einzelnen Felder sowie den Namen der Datei eingeben. Ist das erledigt, erscheint der Hauptbildschirm von Uni-Datei (Bild 3). Die von Ihnen eingegebenen Feldnamen werden links oben angezeigt. In der Kopfzeile des Bildschirms sehen Sie den vorher gewählten Pfadnamen, den Dateinamen, den freien Systemspeicher, die Uhrzeit und das Datum. Darunter stehen die Nummer des angezeigten Datensatzes und die Anzahl der Datensätze, die im Speicher stehen. Am unteren Bildschirmrand ist eine Reihe von »Schaltern«, die sich mit der Maus anklicken lassen. Die Schalter mit den Pfeilen werden verwendet, um den angezeigten Datensatz zu wechseln

– ähnlich wie die Tasten bei einem Kassettenrecorder. In Bild 3 wird gezeigt, wozu Sie welchen Schalter verwenden. Die beiden mittleren Schalter, »Ja« und »Nein«, werden vom Programm bei Abfragen benutzt.

Alle Funktionen der Uni-Datei werden über Menüs mit der Maus ausgewählt. Um Datensätze in Ihre Datei einzugeben, wählen Sie aus dem ersten Menü, »Datei«, den Menüpunkt »Datei pflegen«. Das Programm fragt nun nach den neuen Inhalten für Ihre Datei. Jedes Datenfeld darf höchstens 60 Zeichen lang sein – zu lange Eingaben werden automatisch auf diese Länge gekürzt. Mit Hilfe der Funktionstastenbelegung (wird später erklärt) können Sie sich die Eingabe von oft wiederkehrenden Daten vereinfachen. Die Eingabe wird abgebrochen, indem Sie im ersten Feld eines Datensatzes zuerst <F1> und dann <RETURN> drücken.

Bevor Sie mit Ihrer Datei arbeiten, sollten Sie sie erst einmal speichern. Das geschieht mit einem der »Speichern«-Befehle aus dem Datei-Menü.

Frühzeitig Daten sichern

Wählen Sie »Speichern«, so wird die Datei unter dem Namen gespeichert, den Sie beim Erzeugen der Datei angegeben hatten. »Speichern als« erlaubt Ihnen, einen neuen Namen für die Datei einzugeben. »Speichern privat« speichert die Datei mit einem Paßwort – aber Vorsicht! Der Paßwortschutz ist so einfach, daß Sie dieses Schutzverfahren auf keinen Fall für wirklich geheime Daten verwenden sollten.

Kurz noch zu den weiteren Befehlen des Datei-Menüs: »Datei laden« lädt eine neue Datei. Das Programm erkennt automatisch, ob Sie ihre bisher verwendete Datei gesichert haben und macht eine Sicherheitsabfrage, wenn das nicht so ist. Mit »Datei anhängen« hängen Sie eine Datei an die Datei im Speicher an. Dabei muß die Anzahl der Felder natürlich bei beiden Dateien übereinstimmen. Außerdem darf die maximale Datensatzzahl nicht überschritten werden.

Um eine Datei in einem anderen Verzeichnis auf der Diskette zu verwenden, wählen Sie »Disk-Pfad ändern« und geben den neuen Pfadnamen ein. <RETURN> übernimmt den alten, falls Sie doch nichts ändern möchten. Einen neuen

Sortieren à la carte

Dateinamen geben Sie mit »Dateinamen ändern« ein, und »Directory zeigen« zeigt – wer hätte das gedacht? – das Inhaltsverzeichnis des gewählten Pfades.

Schließlich können Sie mit »Neustart« das Programm von vorne beginnen (beispielsweise wenn Sie sich total verfranzt haben oder um mit einer anderen Datei zu arbeiten) und mit »Ende« das Programm beenden – in beiden Fällen werden alle Daten gelöscht.

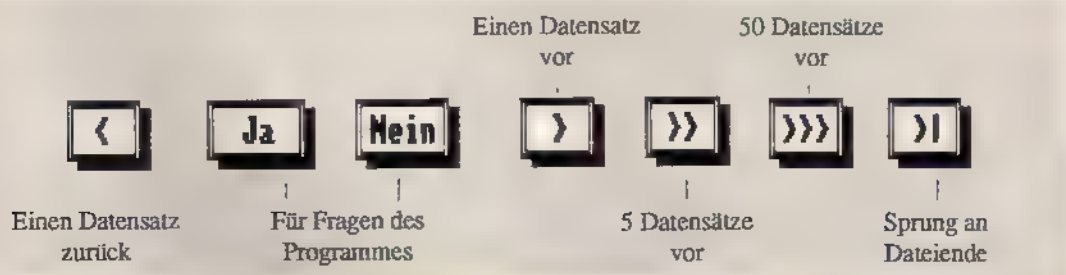
Zurück zu Ihrer Datei. Das »Datensatz«-Menü hilft Ihnen bei der Verwaltung Ihrer Daten. Hier können Sie einen Datensatz löschen, duplizieren und editieren. Diese Befehle beziehen sich immer auf den Datensatz, der gerade auf dem Bildschirm steht. Beim Löschen wird der letzte Datensatz der Datei an die Stelle des gelöschten Datensatzes geschrieben. »Duplizieren« kopiert den aktuellen Datensatz an das Ende der Datei. Um den aktuellen Datensatz zu ändern, wählen Sie »editieren« und geben die neuen Feldinhalte ein – soll ein Feld nicht verändert werden, drücken Sie einfach <RETURN>.

»Gehe zu Datensatz Nummer« bringt den Datensatz mit der eingegebenen Nummer auf den Bildschirm. Um die Feldnamen zu ändern, wählen Sie »Feldnamen editieren« – das Ändern funktioniert ähnlich wie bei »Datensatz editieren«.

Sehr wichtig für eine Dateiverwaltung sind die Sortierfunktionen. Die Uni-Datei kann Ihre Daten nach jedem Feld aufsteigend und absteigend sortieren. Ein Beispiel: Sie wollen Ihre Adreßdatei alphabetisch nach dem Nachnamen sortieren. Dazu wählen Sie »Aufsteigend sortieren«. Es erscheint ein kurzer Hinweis, in dem Sie gefragt werden, ob Sie wirklich sortieren wollen. Klicken Sie auf »Ja«. Jetzt müssen Sie dem Programm sagen, nach welchem Feld es sortieren soll: Dazu wählen Sie aus dem »Suche nach«-Menü den Namen des Feldes aus, nach dem sortiert werden soll. Während des Sortiervorgangs wird grafisch angezeigt, wie groß der schon sortierte Anteil der Datei ist. Das absteigende Sortieren funktioniert nach dem gleichen Muster.

Nachdem Sie jetzt schon einmal in das »Suche nach«-Menü hineingeschnuppert haben, wissen Sie bereits, daß in diesem Menü alle Feldnamen Ihrer Datei stehen. Wenn dieses Menü nicht gerade zum Sortieren »mißbraucht« wird, dient es zum Suchen von Daten. Dazu wählen Sie einfach den Namen des Feldes, nach dem Sie suchen wollen. Die Uni-Datei fragt nun, nach was Sie suchen wollen; geben Sie den Ausdruck an, nach dem Sie suchen. Um hier abzubrechen, tippen Sie wie üblich <F1> und <RETURN>. Beim Suchen gilt folgende Regel: Wenn der Suchbegriff mit dem Anfang des Datenfeldes übereinstimmt, wird der Datensatz angezeigt. Um weiterzusuchen, nachdem ein Datensatz gefunden wurde, wählen Sie »Weitersuchen«.

Was nutzt die beste Dateiverwaltung, wenn man die Daten nicht ausdrucken kann? Nichts. Deshalb sollten Sie sich jetzt einmal das Drucker-Menü der Uni-Datei ansehen. Hier haben Sie zunächst einmal die Möglichkeit, den Datensatz auszudrucken, der gerade auf dem Bildschirm steht. Wählen Sie »Selektierte Datensätze«, so müssen Sie wie beim Suchen einen Feldnamen auswählen und einen Suchbegriff eingeben. Das Programm druckt dann alle Datensätze, die mit diesem Suchkriterium übereinstimmen. Um alle Datensätze



»Knöpfe« am unteren Bildschirmrand. Durch Anklicken wird ihre Funktion aktiviert.

zu drucken, verwenden Sie den Befehl »Alle Datensätze«.

Die Drucker-Parameter lassen sich bei diesem Programm sehr komfortabel einstellen. Dazu wählen Sie »editieren« aus dem »Drucken«-Menü. Es erscheint ein großes Fenster (Bild 4), in dem Sie alle Druckerparameter einstellen. Das geschieht, indem Sie mit der Maus auf die Schalter rechts im Fenster klicken. Die meisten Dinge in diesem Fenster erklären sich von selbst. Bei »Etikettenhöhe« stellen Sie ein, wie viele Felder der Datei untereinander gedruckt werden sollen – stellen Sie z.B. die Etikettenhöhe auf drei, werden nur die ersten drei Felder jedes Datensatzes gedruckt. Setzen Sie den Schalter bei »Informations-Zeile drucken« auf »ja«, wird an den

Daten wie gedruckt

Anfang des Ausdrucks eine Titelseite mit Dateinamen und Datum gesetzt. Interessant ist die Frage »Datenfelder nebeneinander in eine Zeile drucken«. Damit machen Sie Tabellen aus Ihren Daten. Wenn Sie den Schalter dazu auf »Ja« schalten, erscheint ein neues Fenster (Bild 5). Hier stellen Sie ein, wie breit jedes Datenfeld in der Tabelle gedruckt werden soll. Dazu bewegen Sie den blinkenden Stern mit Hilfe der beiden Pfeile zu dem Feldnamen, dessen Breite in der Tabelle Sie ändern wollen. Mit den Rechts- und Linkspleien ändern Sie die Breite – der »+/-«-Schalter macht größere Änderungen. Um ein Feld nicht zu drucken, setzen Sie den

Wert auf Null. Alle Werte werden vom Programm addiert und mit den Abständen zwischen den Tabellenspalten verrechnet. Das Ergebnis dieser Rechnung wird bei »Sie haben eine Zeilenbreite von x eingestellt« angezeigt. Achten Sie darauf, daß Ihr Drucker so breite Zeilen in der eingestellten Schrift drucken kann, sonst gibt es Probleme beim Drucken. Mit einem Klick auf »Ok« gelangen Sie zurück zur Druckerparameter-Einstellung; ein weiterer Klick auf »Ok« in diesem Fenster verläßt die Einstellung der Druckerparameter.

Im »Service«-Menü gibt es etwas sehr praktisches: die Funktionstastenbelegung. Dazu wählen Sie »F2 bis F10 belegen«. Hier werden diese Tasten mit einem beliebigen Text mit bis zu 60 Zeichen Länge belegt. Die F-Tasten-Belegung ist im Menüpunkt »Dateipflegen« und »Datensatz editieren« benutzbar. Die Anweisung »Status« im Service-Menü zeigt die Speicherbelegung des Programmes, und mit »Version« erhalten Sie eine Information über den Autor, Rainer Karwies.

Sollten Sie einmal die Anzahl der Felder Ihrer Datei oder deren Reihenfolge verändern wollen, laden Sie das Programm »UniUtil«. Bevor Sie es anwenden können, müssen Sie eventuell den Pfadnamen am Anfang des Programmes ändern. Ist das Programm korrekt gestartet worden, erscheint der Hinweis »Bitte mit der Maus einen Menüpunkt anwählen«. Um die Anzahl der Felder einer Datei zu erhöhen, wählen Sie den entsprechenden Menüpunkt. Daraufhin wird nach dem Dateinamen gefragt.

Nachdem Sie ihn eingegeben haben, erscheint die Frage »Wie viele Felder soll ich an die x vorhandenen anhängen (Max = x)«. Geben Sie ein, wie viele Felder Sie zusätzlich benötigen. Dabei können Sie maximal die Anzahl angeben, die vorher bei »Max« angezeigt wurde. Nun fragt das Programm nach dem Namen des oder der zusätzlichen Felder und erweitert die Datei entspre-

sen sich jeweils zwei Felder der Datei gegeneinander vertauschen. Dazu geben Sie »J« ein und dann, durch Komma getrennt, die Nummern der zu vertauschenden Felder. Ein Beispiel: Verwenden Sie

1,5

um die Felder 1 und 5 zu vertauschen.

Wie breit darf jedes Datenfeld gedruckt werden ?
(0 = nicht drucken)

Herr/Frau	20
Vorname	20
Name	20
Strasse	20
PLZ	20
Ort	20
Bezirk	20
Vorwahl	20
Telefon	20
Bemerkung 1	20
Bemerkung 2	20
Kundennummer	20
Auftragsnr.	20
Let. Auftrag	20
Kredit	20
Rabatt	20
Skonto	20
Satz OK	20

Sie haben eine Zeilenbreite von 360 Zeichen eingestellt.

Anzahl Spalten zwischen Datenfeldern (in Zeichen) 1

Bild 5. Dieses Fenster verwenden Sie, um aus Ihren Daten Tabellen zu machen.

chend. Zum Speichern der Datei wählen Sie »Geänderte Datei sichern als«.

Das Löschen einzelner Felder funktioniert ähnlich: Sie wählen »Anzahl Datenfelder verringern« und geben wiederum den Dateinamen ein. Nun sehen Sie alle Datenfelder zusammen mit einer Nummer aufgelistet. Um ein bestimmtes Feld zu löschen, geben Sie seine Nummer ein. Achtung: Wenn Sie ein Feld löschen, löschen Sie auch alle Inhalte dieses Feldes. Falls Sie doch nichts löschen wollen, und um das Löschen zu beenden, geben Sie »0« ein. Auch hier müssen Sie »Geänderte Datei sichern als« verwenden, um die neue Datei zu sichern.

Etwas komplizierter wird es, wenn Sie die Reihenfolge der Felder in einer Datei ändern wollen. Dazu müssen Sie zuerst »Anzahl Datenfelder erhöhen« wählen und den Namen Ihrer Datei eingeben. Auf die nun folgende Frage geben Sie »0« ein und drücken <RETURN>. Nun wählen Sie den Menüpunkt »Reihenfolge der Felder ändern«. Das Programm zeigt Ihnen die Felder Ihrer Datei mit Nummern an. Hier las-

Geben Sie nun wieder »J« ein, können Sie wieder zwei Felder vertauschen und so weiter. Mit »N« auf die Frage »Willst Du zwei Felder vertauschen« bricht die Funktion ab; speichern Sie Ihre Datei mit »Geänderte Datei sichern als«.

Wollen Sie die Feldanzahl einer Datei erhöhen oder reduzieren, müssen Sie »Programm neu starten« wählen und diese Operation wie beschrieben ausführen.

Leider gibt es bei einem 512-KByte-Amiga manchmal Speicherprobleme mit »UniUtil«. Es hat sich am besten bewährt, keine anderen Programme gleichzeitig laufen zu lassen und das Programm von der Workbench aus zu starten.

Bereits nach kurzer Einarbeitungszeit werden Sie feststellen, daß Uni-Datei Ihnen die Arbeit sehr einfach macht. Die einzelnen Funktionen sind sauber durchdacht und so schnell, daß Sie nur mit geringen Wartezeiten rechnen müssen. Dazu kommt noch der Vorteil, daß Sie – dank Basic – vollen Zugriff auf alle Funktionen haben und diese nach Ihren Wünschen anpassen können. Also: Legen Sie los. kn

*** Uni-Datei Druckerparameter ***

Linker Rand (in Zeichen)	5	< ± >
Etikettenhöhe (in Zeilen)	18	< ± >
Anzahl Zeilen zwischen Etiketten	1	< ± >
Informations-Zeile drucken		Nein
Zeilenabstand in Zoll		1/6
Druckqualität (NLQ/Draft)		Draft
Schrift (Pica/Elite/schmal)		Pica
Feldnamen mitdrucken		Ja
Datenfelder nebeneinander in eine Zeile drucken		Nein
		OK

Bild 4. Hier werden die Druckerparameter eingestellt

EINGABEHELFE

Checksummer

Dieses Programm ist unentbehrlich beim Abtippen unserer Listings. Es hilft, Tippfehler zu vermeiden und spart viel Zeit.

En längeres Listing ohne Fehler abzutippen ist (fast) unmöglich. Aus diesem Grund haben wir in Ausgabe 3/88 des AMIGA-Magazins eine Eingabehilfe – den Checksummer »Checkie 42« veröffentlicht. Die hier vorgestellte Version 1.1 enthält erweiterte Funktionen und bietet mehr Komfort. Damit möglichst viele unserer Leser dieses Programm auch tatsächlich anwenden, haben wir es möglichst kurz gehalten und in einer Sprache programmiert, die alle Abtipper besitzen: Amiga-Basic.

Die Form der Listings

Die Listingzeilen bestehen aus einer bis zu vierstelligen Zeilennummer, der zwei- bzw. dreistelligen Prüfsumme und der eigentlichen Programmzeile. Beispiel:

```
10 TTQ print "Hallo!"
  |
  | Prüfcode
  |
  | Zeilennummer
```

Nach einer Leerstelle im Anschluß an die Zeilennummer stehen bis zu drei Zeichen Prüfcode. Die einzelnen Zeichen können sein eine Ziffer (»0« bis »9«), ein kleiner Buchstabe (»a« bis »z«) oder ein Großbuchstabe (»A« bis »Z«). Die ersten beiden Zeichen der Prüfsumme sind der eigentliche Prüfcode. Im dritten Zeichen ist die Spaltenposition der ersten »Nicht-Leerstelle« verschlüsselt. Das ist für diejenigen Anwender interessant, welche die Struktur des Listings, also die Einrückungen durch Leerzeichen, übernehmen wollen. Ist dies nicht Ihre Absicht, können Sie die Eingabe der Checksumme schon nach den ersten beiden Zeichen mit <Return> abschließen. Bei Checkie 42 muß die Groß- und Kleinschreibung so wie im Listing abgedruckt übernommen werden.

Der Umgang mit Checkie 42

Nach dem Start fragt das Programm nach einem Dateinamen. Unter dem angegebenen Namen speichert Checkie 42 die eingegebenen Listingzeilen ab. Existiert bereits eine Datei mit diesem Namen auf der Diskette, so haben Sie mit der Abfrage »Nur Checksummer ausgeben?« zwei Möglichkeiten:

- <j> Ausgabe der Datei mit Checksumme auf den Bildschirm oder den Drucker.
- <n> Einlesen der Programmzeilen aus der vorhandenen Datei und Eingabe der Checksumme mit der Tastatur.

Eingabehinweis

Geben Sie »Checkie 42« (Version 1.1) bitte mit Checkie 42 aus der AMIGA-Ausgabe 3/88 oder 12/87 ein. Sollten Sie die alte Eingabehilfe nicht besitzen, so tippen Sie das Listing für die Version 1.1 im normalen Basic-Editor ohne Prüfsummen und Zeilennummern ab.

Beide Alternativen sind gedacht für Anwender, die ein Listing nicht mit dem Zeileneditor des Checkie, sondern mit einem schnelleren und/oder komfortableren Editor ihrer Wahl – z. B. dem Editor von Amiga-Basic (mit »... a« speichern) erfaßt haben.

Checkie 42 errechnet nach der Eingabe <j> die Prüfsummen Ihres Textes und Sie können diese dann mit dem Listing im AMIGA-Magazin vergleichen. Bei der Ausgabe auf den Bildschirm schreibt das Programm die Programmzeilen inklusive Checksummen zusätzlich in eine Datei auf Diskette mit dem Zusatz ».chk«. Diese können Sie später z. B. mit dem CLI-Befehl TYPE ohne erneute Berechnung der Prüfsumme noch einmal ausgeben.

Haben Sie »Nur Checksumme ausgegeben?« mit »n« beantwortet, dann können Sie dem Programm den Vergleich überlassen, in dem die Frage »Eingabe aus Datei« mit »j« beantwortet wird. Dann brauchen Sie nur noch die Checksummen eingeben. Der Checksummer holt sich die Zeile aus der angegebenen Datei statt von der

Tastatur. Entspricht die eingegebene Prüfzahl nicht der errechneten, kann die Zeile gleich korrigiert werden.

Beantworten Sie obige Frage mit »n«, zählt Checkie die in der Datei vorhandenen Zeilen und wartet mit der Zeilennummer »Anzahl+1« auf die Eingabe einer neuen Zeile. Alle weiteren Eingaben hängt das Programm an die bestehende Datei an. Diese Funktion ist sinnvoll, wenn Sie ein Listing in mehreren Teilen abtippen wollen.

So tippen Sie Listings ab

Haben Sie anfangs einen Dateinamen eingegeben oder Sie hängen neue Zeilen an eine bestehende Datei an, dann arbeiten Sie im normalen Eingabemodus. Checkie 42 schlägt dabei eine Zeilennummer vor und wartet auf die Prüfsumme. Nach Eingabe derselben taucht der Cursor zwischen den zwei Trennstrichen auf. Dort muß nun die Zeile »ohne« Zeilennummer und Prüfsumme eingegeben werden. Nach Betätigen der Taste <Return> berechnet Checkie die Prüfsumme. Leerstellen vor und hinter der Programmzeile werden ignoriert. Stimmen Programmzeile und Prüfsumme mit derjenigen im Listing überein, speichert der Checksummer die Eingabe ab und wartet auf die nächste Zeile.

Einfügemodus

Wahrscheinlich wird eine abgetippte Zeile mal einen Fehler enthalten. Checkie 42 positioniert den Cursor dann an den Anfang der Zeile und wartet auf die korrekte Eingabe. Korrekturen lassen sich mit der Backspace- oder Delete-Taste durchführen. Um Zeichenfolgen einzufügen, kann kurzfristig mit <F2> der Einfügemodus eingeschaltet werden. Dieser Modus sollte allerdings nach der Fehlerkorrektur wieder ausgeschaltet werden, da er die Eingabe verlangsamt.

Sonderfall: Prüfsumme ignorieren

Möchten Sie z. B. eine Kommentarzeile nicht »original« übernehmen, läßt sich trotz einer falschen Prüfsumme eine Übernahme der Zeile mit der Funktionstaste <F6> erzwingen. Sie können damit aber auch falsche Programmzeilen übernehmen. Verwenden Sie deshalb die Taste <F6> nicht gewohnheitsmäßig. Der Checksummer teilt Ihnen nach Beenden des Programms mit, wieviel Zeilen er ungeprüft übernommen hat.

Prüfsumme und Zeilennummer ändern

Natürlich kann es auch vorkommen, daß die Programmzeile zwar richtig abgetippt wurde, sich bei der Prüfsumme aber ein Fehler eingeschlichen hat. Nach Betätigen von <F1> kann die Prüfsumme korrigiert werden. Während der Eingabe der Prüfsumme läßt sich mit <F7> die vom Programm vorgeschlagene Zeilennummer verändern. Damit können Sie gezielt nur bestimmte Teile eines Listings übernehmen.

Haben Sie eine mit einem anderen Editor geschriebene Programmdatei überprüft und nur in wenigen Zeilen Fehler festgestellt, lassen sich durch Vorgabe der Nummern diese Zeilen gezielt ändern. Bei Angabe der Zeilennummer in aufsteigender Reihenfolge benötigt das Programm übrigens erheblich weniger Zeit für die Suche der Zeilen in der jeweiligen Datei. Um die versehentliche Übernahme fehlerhafter Zeilen zu verhindern, sperrt das Programm bei fehlender Übereinstimmung der Prüfsummen die Taste <F7> (Änderung der Zeilennummer).

Fehlerfrei abtippen

Eingabe beenden

Die Kombination <Ctrl-e> beendet den Programmablauf nach vollständiger Eingabe des Listings oder für eine Unterbrechung.

Am Schluß noch ein Tip für alle Leser, denen unser Basic-Editor zu langsam ist. Die Berechnung der Prüfsummen erfolgt im Unterprogramm »CalcSumme«. Dieser Teil ist sehr einfach in schnellere Sprachen, wie beispielsweise C, umsetzbar.

Wer schon einmal Fehler in einem abgetippten Listing gesucht hat, der weiß, wie frustrierend diese Arbeit sein kann. Nutzen Sie deshalb den »Checkie 42«. Sie sparen viel Zeit und müssen sich nicht dauernd auf die Suche nach tückischen Fehlern begeben.

Dieter Behlich/kn


```

166 QH COLOR Farbe,0
167 P6 LOCATE sy,sx-15
168 In PRINT "Checksumme:"
169 ni LOCATE sy,sx+1
170 ya e$=INKEY$
171 IT IF e$="" THEN blinken
172 Xp COLOR 1,0
173 d1 e=ASC(e$)
174 FM IF e=5 THEN
175 KU4 FEnde=wahr : REM <CTRL e>
176 hF i=AnsCs2+1 : REM fertig
177 Co2 ELSEIF e=135 THEN
178 Co4 GOSUB NeuZeile
179 Bz e=135
180 LB2 ELSEIF e=8 THEN
181 4a4 IF i>1 THEN
182 s26 i=i-1 : REM <BS>
183 XY LOCATE sy,sx+1 : PRINT "."
184 un4 END IF
185 Tn2 ELSEIF e=13 THEN
186 K24 IF i=AnsCs2 THEN i=AnsCs2+1 : REM <CR>
187 dL2 ELSE
188 ra4 IF e>47 AND e<58 THEN
189 v16 e=e-48 : REM 0-9
190 AI4 ELSEIF e>64 AND e<91 THEN
191 Rv6 e=e-55 : REM A-Z
192 JX4 ELSEIF e>96 AND e<123 THEN
193 Ex6 e=e-61 : REM a-z
194 J84 ELSE
195 yz6 GOTO blinken : REM weder noch
196 6a4 END IF
197 JY PRINT e$;
198 va cs(1)=e
199 D7 i=i+1
200 A32 END IF
201 wn IF i<=AnsCs2 THEN blinken
202 F50 ESEnde:
203 2K2 COLOR 1,0
204 Oh LOCATE sy,sx-15
205 a0 PRINT "Checksumme:"
206 g1 RETURN
207 aE0 NeuZeile:
208 dU2 IF FZok = wahr THEN
209 P44 NeuZeile=0
210 nn WHILE e<>13 OR NeuZeile=0
211 Ux6 LOCATE sy,1:PRINT USING "####":NeuZeile;
212 oF e=ASC(INPUT$(1))
213 gP IF e>47 AND e<58 THEN NeuZeile=NeuZeile*10+e-48
214 Jn IF NeuZeile > 9999 THEN e=8
215 In IF e=8 THEN NeuZeile=INT(NeuZeile/10)
216 vJ4 WEND
217 kh IF Checkfile THEN
218 tQ6 IF NeuZeile < Zeile THEN
219 yy6 WHILE NOT EOF(1)
220 VGA LINE INPUT #1,e$
221 od PRINT #2,e$
222 lp8 WEND
223 MH CLOSE 1 : CLOSE 2
224 FY GOSUB backup
225 Xe OPEN dn$+".bak" FOR INPUT AS #1
226 T1 OPEN dn$ FOR OUTPUT AS #2
227 9t Zeile=1
228 vV6 END IF
229 Yp WHILE (NeuZeile > Zeile) AND (NOT EOF(1))
230 FQ6 LINE INPUT #1,e$
231 nn PRINT #2,e$
232 L7 Zeile=Zeile+1
233 Co6 WEND
234 oR IF EOF(1) THEN
235 bL8 CLOSE 1
236 81 NeuZeile=Zeile
237 uN LOCATE sy,1:PRINT USING "####":NeuZeile;
238 9G Checkfile=0
239 ng6 END IF
240 ch4 END IF
241 vV Zeile=NeuZeile
242 gJ2 END IF
243 H20 RETURN
244 IR EingabeZeile:
245 n72 x=cs(AnsCs2)
246 8W0 weiter:
247 ga2 cy=zy+INT(x/LBZeile):cx=sx+(x MOD LBZeile)
248 vy LOCATE cy,cx
249 o6 COLOR 0,1

```

```

250 nI PRINT CHR$(z(x));
251 y1 LOCATE cy,cx
252 68 IF x>apos THEN apos=x
253 rx IF Checkfile AND FZok THEN
254 v14 IF EOF(1) THEN
255 Q86 Checkfile=0 : CLOSE 1
256 J84 ELSE
257 1x6 e$=INPUT$(1,1)
258 6a4 END IF
259 vV2 ELSE
260 Q24 e$=INKEY$
261 922 END IF
262 gp4 IF e$="" THEN weiter
263 01 COLOR 1,0
264 OW PRINT CHR$(z(x));
265 CF LOCATE cy,cx
266 8G e=ASC(e$)
267 UR2 IF ((e AND 127)<32) OR e=127 THEN Controlcode
268 op IF imode THEN GOSUB insert
269 16 PRINT e$
270 X1 z(x)=e : e=30
271 JDO Controlcode:
272 gP2 IF e=13 OR e=10 THEN
273 LM4 RETURN
274 O92 ELSEIF e=30 THEN
275 2p4 a=1
276 8a2 ELSEIF e=29 THEN
277 O34 a=LBZeile
278 602 ELSEIF e=31 THEN
279 DO4 a=-1
280 Ub2 ELSEIF e=28 THEN
281 8g4 a=-LBZeile
282 9a2 ELSE
283 UX4 GOTO moCrs
284 VP2 END IF
285 h8 x=x+a
286 52 IF x>=0 AND x<LBZeile THEN weiter
287 tD x=x-a
288 HB GOTO weiter
289 K20 moCrs:
290 dn2 IF e=8 THEN
291 1a4 IF x>0 THEN
292 Kx6 x=x-1
293 AM LOCATE zy+INT(x/LBZeile),sx+(x MOD LBZeile)
294 Bb FOR i=x TO apos
295 788 z(1)=z(i-1)
296 Ja PRINT CHR$(z(1));
297 50 IF 1 MOD LBZeile=99 THEN PRINT:PRINT TAB(sx);
298 FV6 NEXT i
299 pr z(apos)=32 : PRINT " "
300 y8 apos=apos-1
301 ng4 END IF
302 H12 ELSEIF e=127 THEN
303 Kx4 FOR i=x TO apos
304 Q86 z(1)=z(i-1)
305 8J PRINT CHR$(z(1));
306 19 IF 1 MOD LBZeile=99 THEN PRINT:PRINT TAB(sx);
307 Oe4 NEXT i
308 jo z(apos)=32 : PRINT " "
309 7a apos=apos-1
310 Vy2 ELSEIF e=129 THEN
311 o14 GOSUB EingabeSumme
312 aC x=cs(AnsCs2)
313 902 ELSEIF e=130 THEN
314 513 imode=imode XOR 1
315 tU9 LOCATE 7,28
316 y23 IF imode THEN
317 IT5 PRINT "aus"
318 JS3 ELSE
319 YV5 PRINT "ein"
320 6a2 END IF
321 Xg2 ELSEIF e=131 THEN
322 L54 GOSUB loeschen

```

Listing. Der verbesserte »Checkie 42«. Bitte mit der ersten Version von »Checkie 42« oder ohne Prüfsummen und Zeilennummern eingeben.

(Fortsetzung) ►

Vorschau

AMIGA-SONDERHEFT 10

AmigaBasic ist eine leistungsfähige Programmiersprache, wenn man sie zu nutzen weiß. Wir bieten Ihnen einen leichten Einstieg in die Welt des Programmierens und zeigen, wie Profis in Basic das letzte Byte aus dem Amiga-Betriebssystem hervorzaubern.

Im Sonderheft 10 erfahren Sie alles über das Einbinden und Ansprechen von System-Routinen und Devices. So finden Sie neben den grundlegenden Kursen auch ausführliche Artikel zu VSprites, Hardware-Sprites, Bobs, Superbitmaps, Double Buffering, Dualplayfield, Copper, Gameport Device und vieles mehr.

Geben Sie Ihren Programmen eine professionelle Note mit einem File-Requester, der auch höchsten Ansprüchen genügt. Sie finden ihn und andere hilfreiche Sub-Programme zum einfachen und problemlosen Einbinden in Ihre Projekte in der nächsten Ausgabe.

Wollen Sie Ihren Basic-Programmen auf die Sprünge helfen? Wir stellen die drei verbreitetsten Compiler bzw. Basic-Dialekte - HisoftBasic, AC Basic und GFA Basic - vor und zeigen, wo ihre Stärken und Schwächen liegen.

10 AMIGA

DM 16,-

SONDERHEFT



Das zehnte Sonderheft liegt ab 28.2. an Ihrem Kiosk.

```

323 JN      x=cs(AnsCs2)
324 7Y2     ELSEIF e=134 THEN
325 bD4     RETURN
326 FR2     ELSEIF e=5 THEN
327 2T4     FEnde=wahr
328 eO      RETURN
329 F82     END IF
330 xR      GOTO weiter
331 IF0     Insert:
332 oJ2     IF apos>x THEN
333 Jo4     FOR i=apos TO x STEP -1
334 Qt6     z(1+i)=z(i)
335 q64     NEXT i
336 FR      z(x)=32
337 Ny      apos=apos+1
338 Kp      IF apos=LBZeile THEN apos=apos-1:s(LBZeile)=32
339 uK      FOR i=x TO apos
340 lI6     PRINT CHR$(z(i));
341 R1      IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(x);
342 xD4     NEXT i
343 yA      LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
344 UN2     END IF
345 vX      RETURN
346 aX0     CalcSumme:
347 3s2     a=0 : b=0 : c=0
348 a2      IF e=134 THEN
349 pv4     FZok=wahr
350 WX      FF6=FF6+1
351 Ga2     ELSE
352 hv4     WHILE z(apos)=32 AND apos>0
353 pI6     apos=apos-1
354 9x4     WEND
355 N1      IF apos>0 THEN
356 O36     WHILE z(c)=32
357 Bt8     c=c+1
358 D16     WEND
359 Jo4     END IF
360 EJ      FOR i=c TO apos
361 Y26     j=(1-c) MOD AnsFak

```

```

362 Qs      k=(1+1-c) MOD AnsFak
363 e1       a=a+((z(1) AND 127)-32)*Faktor(1)
364 2E      b=b+((z(1) AND 127)-32)*Faktor(k)
365 3a4     NEXT i
366 pA      cs(4)=a+Zeile-INT((a+Zeile)/62)*62
367 4N      cs(5)=b+Zeile-INT((b+Zeile)/62)*62
368 4Q      FZok=(cs(1)=cs(4)) AND (cs(2)=cs(5))
369 tm2     END IF
370 Ev      RETURN
371 pM0     Uebernahme:
372 xD2     FOR i=0 TO apos
373 lB4     PRINT#2,CHR$(e(i));
374 Tj2     NEXT i
375 v24     PRINT#2,""
376 9R2     Zeile=Zeile+1
377 R3      RETURN
378 W90     fertig:
379 MJ2     IF Checkfile THEN
380 Z24     WHILE NOT EOF(1)
381 6r6     LINE INPUT#1,e$
382 DE      PRINT#2,e$
383 cQ4     WEND
384 Ok      CLOSE 1
385 922     END IF
386 5q      CLOSE 2
387 nt      CLS
388 Fg      LOCATE 12,35
389 WL      PRINT "F E R T I G !!!"
390 I9      LOCATE 20,1
391 D1      IF FF6<>0 THEN
392 C14     PRINT "ACHTUNG!!! ";
393 9H      PRINT FF6;" Zeile(n) wurde(n) ungeprüft gespeichert."
394 lB2     END IF
395 J1      RETURN
396 1988    MAT

```

Listing. Der verbesserte »Checkie 42« (Schluß)

Stoff für Ihren Amiga



HiSoft-Basic-Compiler

Mit HiSoft Basic gibt es endlich auch einen Basic-Compiler für den Amiga. Der interaktive Editor-, Kompilier- und Laufzeitzyklus entspricht dem eines Interpreters.

Der integrierte Editor erlaubt eine komfortable Eingabe. HiSoft Basic unterstützt die Eigenschaften des Amiga mit Fenstern, Grafik-Kommandos, Sprite-Handling und Maschinenzugriffen während des Gebrauchs von Bibliotheken. Es ist voll kompatibel mit Amiga-Basic, andere Standarddialekte für den PC, wie z.B. Microsoft QuickBasic, und den Atari ST können mit geringen Modifikationen kompiliert werden. Rekursive

Unterprogramme und Funktionen sind möglich. Eine Anzahl strukturierter Ausdrücke wie z.B. WHILE...WEND, DO...LOOP UNTIL und SELECT...CASE lassen Sie jede Programmierhürde meistern. Die Größe von Variablen ist nicht beschränkt. Aussagekräftige Fehlermeldungen und Korrekturmöglichkeiten tragen zur komfortablen Handhabung bei.

Hardware-Anforderungen: Amiga 500, 1000 oder 2000 mit Kickstart 1.2 oder höher, 3 1/2"-Diskette, Bestell-Nr. 54127

DM 179,-*
(sFr 161,- /öS 1790,-*)

Devpac Assembler 2.0

Devpac 2.0 ist ein Entwicklungspaket für den Amiga mit komfortablem Editor/Assembler, symbolischem Debugger und Linker zum Einbinden von Hochsprache-Modulen.

GenAm ist ein 68000er-Makroassembler mit integriertem Bildschirmeditor, der bis zu 75.000 Zeilen pro Minute assemblieren kann. Der 2-Paß-Assembler erzeugt sowohl linkbaren als auch direkt ausführbaren Code. Er unterstützt lokale Labels, die Signifikanz beträgt bis zu 127 Zeichen.

Makros können bis zu 36 Parameter beinhalten und – Rekursion inbegriffen – so tief verschachtelt werden, wie Speicherplatz vorhanden ist. MonAm, der Debugger, erlaubt das Setzen von Breakpoints, das Disassemblieren auf Diskette und noch weitere Features, die das Debuggen zum Vergnügen machen.

Hardware-Anforderungen: Amiga 500, 1000 oder 2000 mit mindestens 512 Kbyte, ein Diskettenlaufwerk, 3 1/2"-Diskette, Bestell-Nr. 54131

DM 149,-*
(sFr 135,- /öS 1490,-*)

Devpac-Assembler 2.0 im Test! Amiga-Magazin 4/89: Eines der besten Programme seiner Art für den Amiga.

SuperED C

Multitasking-fähiger Editor als Programmierungsumgebung für den Aztec-C-Compiler (V 3.6). Bestell-Nr. 54139

DM 39,-*
(sFr 35,- /öS 390,-*)

* Unverbindliche Preisempfehlung

INFO-COUPON

Bitte senden Sie mir weiteres Informationsmaterial zu
☐ Devpac-Assembler ☐ HiSoft-Basic-Compiler

Name _____

Straße _____

PLZ/Ort _____

Bitte ausschneiden und einsenden an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Ami SH 9

221 1007




Markt&Technik
 Zeitschriften · Bücher
 Software · Schulung

WWW.HOMECOMPUTERWORLD.COM

AMIGA

BOOK- WARE

Profi-Software unter 100 DM



R. Arbing, I. Krüger
Scriptum

Das leistungsfähige Textverarbeitungsprogramm für den Amiga. Dazu eine ausführliche Programmbeschreibung im Buch, die bei allen Fragen weiterhilft. 1989, 141 Seiten, inkl. Programmdiskette. ISBN 3-89090-650-8
DM 79,-* (sFr 72,70*/öS 672,-*)

Precision Software
Amiga Superbase

Die Einsteiger-Datenbank. Relationales System zum Verwalten von Daten aller Art, auch von IFF-Grafiken; mit integrierter Dideshow-Funktion. 1989, 176 Seiten, inkl. Programmdiskette. ISBN 3-89090-791-1
DM 89,-* (sFr 81,90*/öS 757,-*)

Atlantis **AmigaCall**
Treten Sie ein in die faszinierende Welt der Datenfernübertragung.

Kommunizieren Sie über Mailboxen mit erfahrenen Computer-Anwendern, die Ihnen bei Ihren Problemen weiterhelfen können, oder Sie erhalten auf diesem Wege leistungsfähige Public-Domain-Software.

1988, 133 Seiten, inklusive Programmdiskette. ISBN 3-89090-716-4
DM 99,-* (sFr 91,-*/öS 842,-*)

C. Fuchs **Reflections**

Traumwelt und Realismus - Bilder im Raytracing-Verfahren auf dem Amiga. Dazu im Buch eine ausführliche Bedienungsanleitung. 1989, 156 Seiten, inklusive Programmdiskette. ISBN 3-89090-727-X
DM 98,-* (sFr 90,20*, öS 834,-*)

Atlantis **Trickstudio A**

Animationen in Bild und Ton. Ein Programm zum einfachen Erstellen und Abspielen von bewegten Bildern, die synchron mit Sound unterlegt werden können. Dazu eine ausführliche Dokumentation für die effektive Anwendung. 1988, 86 Seiten, inklusive Programmdiskette. ISBN 3-89090-715-6
DM 99,-* (sFr 91,-*/öS 842,-*)

H. Knappe **Amiga Sounder**

Der Amiga Sounder ist ein Komplett-paket für den Einstieg in die Welt der digitalen Klänge. 1989, 336 Seiten, inkl. 2 Programmdisketten. ISBN 3-89090-709-1
DM 98,-* (sFr 90,20*/öS 834,-*)

In Vorbereitung:

Dr. Glaeser/T. Grohser
3-D-Sprinter Amiga, Version 1.3
interaktive Echtzeit-Animation
Lieferbar 4. Quartal 1989
ca. 250 Seiten, inkl. Diskette
ISBN 3-89090-109-3
ca. DM 98,-* (sFr 90,20*/öS 834,-*)

In Vorbereitung
N. Wirsing

Amiga Audio Entwickler-Paket
Dieses Buch macht Sie zum perfekten Amiga-Tontechniker. Lieferbar 1. Quartal '90, ca. 400 Seiten, inkl. 2 Programmdisketten. ISBN 3-89090-765-2
ca. DM 98,-* (sFr 90,20*/öS 834,-*)

*Unverbindliche Preisempfehlung

INFO-COUPON

Bitte senden Sie mir Ihr Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software.

Name _____

Straße _____

PLZ/Ort _____

Bitte ausschneiden und einsenden an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Ami SH 9


Markt & Technik
Zeitschriften · Bücher
Software · Schulung

WWW.HOMECOMPUTERWORLD.COM

Markt & Technik-Bücher und -Software erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler.